

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

PROJE TAKİP YAZILIMINDA YAPAY ZEKA TABANLI DUYGU
ANALİZİ

MERVE EMEN

KOCAELİ 2025

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**PROJE TAKİP YAZILIMINDA YAPAY ZEKA TABANLI DUYGU
ANALİZİ**

MERVE EMEN

Doç. Dr. Mehmet Zeki KONYAR
Danışman, Kocaeli Üniversitesi

.....

Dr. Öğr. Üyesi Kaplan KAPLAN
Jüri Üyesi, Kocaeli Üniversitesi

.....

Doç. Dr. Yılmaz KAYA
Jüri Üyesi, Batman Üniversitesi

.....

Tezin Savunulduğu Tarih: 23.01.2025

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- Bu tezin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı,

beyan ederim.

☒ Bu tez çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☐ Bu tez çalışması kapsamında üretilen veri ve bilgiler tarafından no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Merve EMEN

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.

Tezin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan **“Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge”** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- ☐ Enstitü yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
- ☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
- ☒ Tezim ile ilgili gizlilik kararı verilmemiştir.

Merve EMEN

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, proje takip yazılımı olan Jira'daki yorumlara duygu analizi uygulanarak kapsamlı bir çalışma gerçekleştirilmiştir. Çalışmanın amacı, Jira üzerinden müşteri memnuniyetini duygu analizi yöntemi ile kolayca belirlemektir. Bu sayede yorumların olumlu, olumsuz ve nötr olduğu hızlı ve maliyet etkin bir şekilde tespit edilebilir.

Tez çalışmamda bilgilerini ve desteğini benden esirgemeyen kıymetli danışman hocam Doç. Dr. Mehmet Zeki KONYAR'a teşekkür ederim.

Hayatımın her anında maddi manevi yanımda olup beni destekleyen, cesaretlendiren ve beni bu günlere getiren annem Fatma KUL ve babam Kazım KUL'a teşekkürü borç bilirim.

Benimle her zaman gurur duyan, yanımda olan ve tez çalışmamda bana destek veren değerli eşim Ömer EMEN'e çok teşekkür ederim.

Ocak – 2025

Merve EMEN

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ.....	i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI	ii
ÖNSÖZ VE TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ	v
TABLOLAR DİZİNİ.....	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	3
3. JIRA.....	7
3.1. Jira Software Sürümü.....	9
3.2. Jira Service Management Sürümü	11
4. MATERYAL VE METOT.....	12
4.1. Makine Öğrenmesi.....	12
4.1.1. Makine Öğrenmesi Yöntemleri	12
4.1.2. Makine Öğrenmesi Süreci	15
4.1.3. Sınıflandırma Algoritmaları	17
4.1.4. Mesafe Ölçüm Metrikleri	22
4.2. Doğal Dil İşleme	24
4.2.1. Doğal dil işleme ön hazırlık adımları	24
4.2.2. Duygu Analizi	27
4.2.3. Metin Vektörizasyonu	28
4.2.4. Dönüştürücüler (Transformers)	31
4.2.5. BERT	32
4.2.6. Öz Dikkat Mekanizması (Self Attention).....	34
4.3. Performans Ölçüm Metrikleri	37
5. DENEYSEL ÇALIŞMALAR	40
5.1. Veri seti.....	41
5.2. Veri Ön İşleme.....	42
5.3. Hiperparametre Optimizasyonu	44
6. DENEYSEL SONUÇLAR.....	46
7. SONUÇLAR VE ÖNERİLER	66
KAYNAKLAR.....	69
KİŞİSEL YAYIN VE ESERLER.....	72
ÖZGEÇMİŞ.....	73

ŞEKİLLER DİZİNİ

Şekil 3.1.	Jira’da açılmış olay kaydı örneği	8
Şekil 3.2.	Jira filtreleme örneği	8
Şekil 3.3.	Jira proje şablon seçimi	10
Şekil 3.4.	İş listesi Oluşturma Örneği.....	10
Şekil 3.5.	İterasyon Planlama	10
Şekil 3.6.	Jira ‘da açılmış bir olay içeriği	11
Şekil 4.1.	DVM modeli çalışma yapısı.....	18
Şekil 4.2.	KA modeli örnek senaryosu	19
Şekil 4.3.	RO modeli çalışma yapısı	20
Şekil 4.4.	EYK modeli çalışma yapısı.....	22
Şekil 4.5.	Gövdeleme işlemi örnek kod parçası	26
Şekil 4.6.	Dönüştürücü mimarisi (Karaca ve Aydın , 2024)	32
Şekil 4.7.	BERT akış şeması	34
Şekil 4.8.	İkili sınıflandırma için karışıklık matrisi yapısı	39
Şekil 4.9.	ROC analizi örneği.....	39
Şekil 5.1.	Önerilen makine öğrenmesi temelli yöntemin akış şeması	40
Şekil 5.2.	Önerilen BERT ile hibrit modelleme temelli yöntemin akış şeması.....	41
Şekil 5.3.	Veri ön işleme akış şeması	43
Şekil 5.4.	TF-IDF uygulanması sonucu veri setinin görünümü	44
Şekil 6.1.	Count vectorizer yöntemi XGBoost modeli karışıklık matrisi.....	48
Şekil 6.2.	Count vectorizer yöntemi DVM modeli karışıklık matrisi	48
Şekil 6.3.	Count vectorizer yöntemi RO modeli karışıklık matrisi	49
Şekil 6.4.	Count vectorizer yöntemi KA modeli karışıklık matrisi	49
Şekil 6.5.	Count vectorizer yöntemi EYK modeli karışıklık matrisi	49
Şekil 6.6.	Count vectorizer yöntemi GNB modeli karışıklık matrisi	50
Şekil 6.7.	Count vectorizer yöntemi MNB modeli karışıklık matrisi.....	50
Şekil 6.8.	Count vectorizer yöntemi XGBoost modeli ROC analizi	51
Şekil 6.9.	Count vectorizer yöntemi DVM modeli ROC analizi.....	51
Şekil 6.10.	Count vectorizer yöntemi RO modeli ROC analizi	52
Şekil 6.11.	Count vectorizer yöntemi KA modeli ROC analizi	52
Şekil 6.12.	Count vectorizer yöntemi EYK modeli ROC analizi.....	52
Şekil 6.13.	Count vectorizer yöntemi GNB modeli ROC analizi.....	53
Şekil 6.14.	Count vectorizer yöntemi MNB modeli ROC analizi	53
Şekil 6.15.	TF-IDF yöntemi XGBoost modeli karışıklık matrisi.....	56
Şekil 6.16.	TF-IDF yöntemi DVM modeli karışıklık matrisi.....	56
Şekil 6.17.	TF-IDF yöntemi RO modeli karışıklık matrisi	57
Şekil 6.18.	TF-IDF yöntemi KA modeli karışıklık matrisi	57
Şekil 6.19.	TF-IDF yöntemi EYK modeli karışıklık matrisi	57
Şekil 6.20.	TF-IDF yöntemi GNB modeli karışıklık matrisi.....	58
Şekil 6.21.	TF-IDF yöntemi MNB modeli karışıklık matrisi	58
Şekil 6.22.	TF-IDF yöntemi XGBoost modeli ROC analizi	59
Şekil 6.23.	TF-IDF yöntemi DVM modeli ROC analizi	59
Şekil 6.24.	TF-IDF yöntemi KA modeli ROC analizi.....	60
Şekil 6.25.	TF-IDF yöntemi RO modeli ROC analizi	60
Şekil 6.26.	TF-IDF yöntemi EYK modeli ROC analizi	60
Şekil 6.27.	TF-IDF yöntemi GNB modeli ROC analizi	61

Şekil 6.28. TF-IDF yöntemi MNB modeli ROC analizi	61
Şekil 6.29. BERT+DVM modeli karışıklık matrisi	63
Şekil 6.30. BERT+RO modeli karışıklık matrisi.....	64
Şekil 6.31. BERT+GNB modeli karışıklık matrisi.....	64
Şekil 6.32. BERT+DVM modeli ROC analizi	65
Şekil 6.33. BERT+RO ROC analizi	65
Şekil 6.34. BERT+GNB ROC analizi	65



TABLÖLAR DİZİNİ

Tablo 4.1. Hamming mesafe hesaplama.....	24
Tablo 4.2. Kelime frekansı çıkarımı.....	29
Tablo 4.3. TF değeri hesabı.....	29
Tablo 4.4. IDF değeri hesabı.....	30
Tablo 4.5. TF-IDF hesabı.....	30
Tablo 4.6. Count vectorizer hesaplama örneği.....	31
Tablo 4.7. Öz dikkat mekanizması sorgu,anahtar, değer matrisi.....	35
Tablo 5.1. Veri setinde bulunan pozitif sınıfına ait öznitelik örneği.....	42
Tablo 5.2. Veri setinde bulunan negatif sınıfına ait öznitelik örneği.....	42
Tablo 5.3. Veri setinde bulunan nötr sınıfına ait öznitelik örneği.....	42
Tablo 5.4. Hiper parameter optimizasyonu sonucu en iyi parametreler.....	45
Tablo 6.1. Count vectorizer değerlendirme metrikleri.....	46
Tablo 6.2. TF-IDF değerlendirme metrikleri.....	54
Tablo 6.3. En iyi parametreler kullanıldığında TF-IDF performans metrikleri.....	62
Tablo 6.4. BERT ile hibrit çalışan modellerin performans metrikleri.....	62
Tablo 7.1. Tez ile diğer çalışmaların karşılaştırılması.....	68

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

Acc	: Accuracy (Doğruluk)
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
BERT	: Bidirectional Encoder Representations From Transformers
DVM	: Destek Vektör Makinesi (Support Vector Machine)
EYK	: En Yakın Komşu (K-Nearest Neighbor)
FN	: False Negative (Yanlış Negatif)
FP	: False Positive (Yanlış Pozitif)
GNB	: Gaussian Naive Bayes (Gaus Naif Bayes)
HOMER	: Hierarchy of Multi-label Classifiers (Çok Etiketli Sınıflandırıcıların Hiyerarşisi)
IDF	: Inverse Document Frequency (Ters Doküman Frekansı)
KA	: Karar Ağacı (Decision Tree)
LR	: Logistic Regression (Lojistik Regresyon)
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
MLP	: Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
MNB	: Multinomial Naive Bayes (Çok Terimli Naif Bayes)
NRC	: National Research Council of Canada (Kanada Ulusal Araştırma Konseyi)
NBTree	: Naive Bayes Tree (Naif Bayes Ağacı)
Pre	: Precision (Kesinlik)
RHN	: Recurrent Highway Network (Tekrarlayan Otoyol Ağı)
RAkEL	: Random k-Labelsets (Rastgele k-etiket Kümeleri)
Rec	: Recall (Duyarlılık)
RO	: Rastgele Orman (Random Forest)
ROC	: Receiver Operating Characteristic (Alıcı İşletim Karakteristiği)
SA	: Standardized Accuracy (Standartlaştırılmış Doğruluk)
TF	: Term Frequency (Terim Frekansı)
TN	: True Negative (Doğru Negatif)
TP	: True Positive (Doğru Pozitif)
XGBoost	: eXtreme Gradient Boosting (Ekstrem Gradyan Artırma)
YSA	: Yapay Sinir Ağları (Artificial Neural Networks)

PROJE TAKİP YAZILIMINDA YAPAY ZEKA TABANLI DUYGU ANALİZİ

ÖZET

Bu tez çalışmasında proje takip yazılımı olan Jira yorumları üzerinde duygu analizi yöntemi ile olumlu, olumsuz ve nötr ifadelerin tahminleme işlemi gerçekleştirilmiştir. Proje takip yazılımları projelerin planlanması, yürütülmesi ve işlerin izlemesini kolaylaştıran bir yazılımdır. Çalışmanın amacı, şirketlerin proje takip yazılımlarında duygu analizi yöntemi kullanarak hızlı ve daha az maliyetli bir şekilde müşteri memnuniyetini belirlemeleridir.

Bu çalışmada Kaggle platformundan elde edilen Jira yorumlarının bulunduğu ve pozitif, negatif, nötr olarak etiketlenmiş 5869 satırlık veri seti kullanılmıştır. Veri ön işleme aşamaları yapıldıktan sonra TF-IDF ve count vectorizer vektörizasyon teknikleri kullanılarak makine öğrenmesi algoritmaları olan Destek vektör makineleri (DVM), Karar ağacı (KA), Rastgele orman (RO), Ekstrem gradyan artırma (XGBoost), K-en yakın komşu (EYK), Multinomial naive bayes (MNB) ve Gaussian naive bayes (GNB) ile modelleme işlemi yapılmıştır. Duygu analizinde en çok kullanılan yöntemlerden biri olan BERT modeli uygulanmış ve BERT modeli ile makine öğrenmesi yöntemlerinden DVM, GNB ve RO algoritmaları hibrit bir şekilde çalıştırılmıştır. K-En Yakın Komşular, XGBoost, RO ve DVM algoritmaları için Gridsearch ile hiperparametre optimizasyonu yapılmıştır. Doğruluk, kesinlik, duyarlılık, F1 puanı, hata matrisi, ROC analizi metrikleriyle modellerin performansları değerlendirilmiştir. Deneysel sonuçlara göre en yüksek sınıflandırma doğruluğu count vectorizer yönteminde XGBoost algoritması ile %86,4 olarak elde edilmiştir. Count vectorizer yöntemi algoritmalarının ortalama doğruluğu %76,2 TF-IDF yöntemi algoritmalarının ortalama doğruluğu %76,9 Gridsearch yöntemi algoritmalarının ortalama doğruluğu %77,7, BERT modeli ve BERT modeliyle hibrit çalışan DVM, GNB ve RO algoritmalarının ortalama doğruluk değeri ise %79,4 olarak tespit edilmiştir.

Anahtar Kelimeler: BERT, Duygu Sınıflandırma, Jira, Makine Öğrenmesi, Proje Takip Yazılımı.

ARTIFICIAL INTELLIGENCE BASED SENTIMENT ANALYSIS IN PROJECT TRACKING SOFTWARE

ABSTRACT

This thesis study aims to predict positive, negative, and neutral expressions using sentiment analysis on comments from the project tracking software, Jira. Project tracking software facilitates project planning, execution, and monitoring of tasks. The objective of this study is to enable companies to determine customer satisfaction in project tracking software quickly and cost-effectively by using sentiment analysis techniques.

In this study, a dataset containing Jira comments obtained from the Kaggle platform, labeled as positive, negative, or neutral, consisting of 5,869 rows, was used. After performing data preprocessing steps, vectorization techniques such as TF-IDF and Count Vectorizer were applied, and machine learning algorithms, including Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), Multinomial Naive Bayes (MNB), and Gaussian Naive Bayes (GNB), were utilized for modeling. The BERT model, one of the most commonly used methods in sentiment analysis, was applied, and hybrid models combining BERT with SVM, GNB, and RF were implemented. Hyperparameter optimization was conducted using the GridSearch method for K-Nearest Neighbors, XGBoost, RF, and SVM algorithms. The performance of the models was evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC analysis metrics.

According to experimental results, the highest classification accuracy was achieved using the Count Vectorizer method with the XGBoost algorithm, reaching 86.4%. The average accuracy of algorithms using the Count Vectorizer method was found to be 76.2%, while the TF-IDF method yielded an average accuracy of 76.9%. The GridSearch method improved the average accuracy to 77.7%, and the BERT model, along with its hybrid combinations with SVM, GNB, and RF, achieved an average accuracy of 79.4%.

Keywords: BERT, Sentiment Analysis, Jira, Machine Learning, Project Tracking Software.

1. GİRİŞ

Doğal dil işleme; metin özetlemede, duygu analizinde ve dil çevirisinde birçok sorun için sıklıkla kullanılan aynı zamanda en çok ihtiyaç duyulan bir alandır. Günümüz çağında alışveriş sitesi yorumları, sosyal medya yorumları ve paylaşımları gibi alanlarda insanlar duygu ve düşüncelerini rahatlıkla paylaşabildikleri için duygu analizi kullanımı artmıştır. İnsanların durumlara göre öfkeli, mutlu veya üzgün olduğu durumlar duygu analizi ile belirlenebilmektedir bununla birlikte bir metnin olumlu mu yoksa olumsuz mu olma durumu duygu analizi ile kolaylıkla tespit edilebilmektedir. Duygu analizi bu gibi durumlar sayesinde işletmecilerin, araştırmacıların en çok kullandığı alan haline gelmiştir. Duygu analizi yöntemleri, derin öğrenme ve yapay sinir ağları gibi modeller sayesinde giderek geliştiği için modellerin performansları daha başarılı hale gelmiş ve analiz sonuçları daha başarılı olmaktadır. Gelişen duygu analizi ilerleyen dönemlerde karşılaşılan zorluklar ortaya çıktıkça daha da gelişip uygulanacak alan sayısı artacaktır. Dijitalleşmenin artmasıyla şirketler de bu sistemden faydalanabilirler. Bu çalışmada iş takip araçlarında biri olan Jira yorumları için duygu analizi yöntemi uygulanmıştır.

Jira, Atlassian firması tarafından kurulan proje ve sorun takip aracıdır (URL-1). Şirketler bu araç sayesinde iş akışlarını kolaylıkla takip etme fırsatı bulurlar. Jira bir sorunun giderilmesi, yeni bir gereksinim veya var olan projenin geliştirilmesi gibi durumlara ihtiyaç duyulabilir. Bu durumlarda proje yönetmeyi daha kolay hale getirmek için proje takip aracı kullanmak gerekmektedir. Jira çevik yazılım sürecini de desteklediği için proje yönetmeyi de kolaylaştırır. Jira’da iş listesi (backlog) oluşturulup iterasyon (sprint) planlaması yapıldıktan sonra yazılım geliştiricilere talep ataması yapılır. Yazılım destek veya saha destek personelleri ise anlık olan hataları çözmeleri için onlara olay kaydı atanmaktadır. Jira’da çalışan personele her atanan görev çözümlenip kapatıldıktan sonra görevi açan kişi çözümü beğendiği veya beğenmediği durumlarda yorum olarak çözümü beğenip beğenmediğine veya isteğini karşılayıp karşılamadığına dair düşüncelerini yazabilir. Örneğin iş kısa sürede çözülmüş ise ve bu durumdan memnun kalır sonrasında teşekkür eder bu yorum ise olumlu olarak algılanır.

Bu çalışmadaki amaç, Jira uygulamasında bulunan olaylar için yapılan her bir yorumun olumlu, olumsuz ve nötr olarak değerlendirmesini duygu analizi yöntemi ile yapmaktır.

Bu sayede aynı konu üzerinde olumsuz bir geri bildirim olduğu fark edilip firmanın hangi alanda eksiğinin olduğu da belirlenebilir. Firmalar yıllık olarak çalışanlarını değerlendirmek istediklerinde veya prim verilmesi gibi durumlarda tüm yorumlar tek tek okunup değerlendirmek zaman alıcı olur ve o kişi özelinde genellikle iyi veya kötü çalıştığını belirlemek kesinlik kazandırmaz. Bu çalışma sayesinde ekip içi iletişim ve hız konusunda artışa katkı sağlayacaktır. Geri bildirimlerin durumuna göre işletmeler daha proaktif çözümler sunabileceklerdir. Olumlu sonuç alan ekip üyeleri ise geri bildirimlerden memnun kalıp motivasyon açısından daha verimli çalışabileceklerdir.



2. LİTERATÜR TARAMASI

Proje takip yazılımları projelerin yürütülmesi ve yönetilmesi için kullanılan önemli bir araç haline gelmiştir. Bu araçların sadece proje yönetilmesi değil aynı zamanda müşteri ve kullanıcıların geri bildirimleri projeler için önem arz etmektedir. Bu çalışmada da bu durumu ele almak için duygu analizi yöntemi kullanılmıştır. Günümüzde de duygu analizi yöntemi kullanımı artmış ve birçok kullanılmaya başlanmıştır. Proje takip yazılımlarında da duygu analizi yöntemine literatürde birçok çalışmada yer verildiği gözlemlenmiştir.

(Cabrera-Diego ve diğ., 2020) çalışmasında Stack overflow ve Jira yazılım geliştirme platformlarındaki yorumlardan elde edilen veri setleri kullanarak metin madenciliği ile duygu sınıflandırmasını ele almışlardır. Yöntem olarak Hierarchy of Multi-label Classifiers (HOMER) ve Random k Labelsets (RAkEL) çoklu etiketli sınıflandırıcıları kullanmışlardır. Homer etiketleri hiyerarşik gruplar olarak organize eder ve her grup için ayrı sınıflandırıcılar kullanır. Rakel ise rastgele etiket setleri oluşturur ve bu setler üzerinde sınıflandırıcıları eğitir. Çok etiketli polarite olarak SenticNet ve NRC (National Research Council of Canada) kullanılmıştır. SenticNet ile kelimelerin duygusal içeriklerini ve kavramsal ilişkileri ele alınmıştır. NRC ile ilişkili duygular (öfke, korku, şaşkınlık, üzüntü ve neşe) ve kutuplaşma (olumlu, olumsuz) kaynak kullanılarak etiketleme yapılmıştır. Ayrıca, sınıflandırıcıların performansı mevcut en iyi araç olan EmoTxt ile karşılaştırılmıştır. Emotxt metinlerdeki olumlu, olumsuz ve nötr durumlarını analiz etmek için kullanılan bir modeldir. Bir metnin birden fazla duygusal etikete sahip olabileceği üzerine düşünülerek çalışma ilerletilmiştir. Örneğin bir metinde hayal kırıklığı heyecan gibi duygular bir arada bulunabilir. Bu nedenle çok etiketli yaklaşımı önermektedir. Bu bakış açısı ile çalışıldığında daha doğru sonuçlar elde edilebilir. Rakel yöntemine göre Jira veri setinde temel kök haline dönüştürme işlemi sonrası F1 puanı %55,1 kök haline dönüştürülmediğinde ise %58,9 olduğu tespit edilmiştir. Homer yöntemine göre ise Jira veri setinde temel kök haline dönüştürme işlemi sonrası F1 puanı %57,1 dönüştürülmediğinde ise %58,5 olarak bulunmuştur. Emotxt yönteminde Jira veri setinde temel kök haline dönüştürülme yapılmadığında F1 puanı %48,8 olarak bulunmuştur. HOMER ve RAkEL, EmoTxt'e kıyasla Jira yorumları üzerinde daha iyi sonuçlar verdiği gözlemlenmiştir.

(Choetkiertikul ve diğ., 2017) çalışmasında Jira'daki 8 açık kaynaklı projeden elde edilen veriler kullanılarak gecikme olasılığı ve gecikmenin süre tahmini ele alınmıştır. Bu araştırma yazılım projelerinin zamanında tamamlanmasına yönelik bir çalışmadır. Teslim tarihine göre gecikme süresini tahminlemeyi amaçlar. Projenin tamamlanma süresinin tahmin edilmesi projenin etkili şekilde yönetilmesi açısından kolaylık sağlar. Veri setinde tartışma süresi, type (story, task, bug), bir sorunun yeniden açılma sayısı ve önceliği ele alınmıştır. Tahminleme için rastgele orman (RO), sinir ağları, karar ağacı (KA), naive bayes (NB), naive bayes tree (NBTree), derin sinir ağı, gradyan artırma makineleri kullanılmış ve en iyi kesinlik değeri %96 sonucunu veren RO algoritması olduğu gözlemlenmiştir.

(Choetkiertikul ve diğ., 2019) çalışmasında bir işin gereksinimlerini belirlemek ve projenin karmaşıklığını değerlendirmek için hikaye puanlarının tahminlerini geliştirir. Hikaye puanı yazılım projesinin karmaşıklığını ve iş yükünü temsil eder. Bu tahmin ile proje planlaması daha kolaylaşır hale gelmektedir. Jira'nın REST API'sinden Apache, Appcelerator, DuraSpace, Atlassian, Moodle, Lsstcorp, Mulesoft, Spring ve Talendforge açık kaynak kodlu projeler çekilmiştir. Veri setinde 23313 sorun ve hikaye bulunmaktadır. 16 farklı projeden elde edilmiştir. Yaklaşımın temel yeniliği, iki güçlü derin öğrenme mimarisinin birleşiminde yatmaktadır: uzun kısa süreli bellek (LSTM) ve tekrarlayan otayol ağı (RHN). Bu yaklaşımı Deep learning model for Story point Estimation (Deep-SE) olarak adlandırılmıştır. Değerlendirme sonucu olarak MAE (Ortalama mutlak hata) ve Standartlaştırılmış Doğruluk (SA) bakılmıştır. Bu çalışmada Jira software veri setinde LSTM+SVM ve LSTM+ATLM (Automatically Transformed Linear Model) modellerinin SA oranları sırasıyla %41,97 %45,18 şeklinde performans sergilemiştir. Tahmine dayalı performans MAE'yi azaltarak veya SA'yı artırarak iyileştirilebilir sonucuna varıldığı gözlemlenmiştir.

(Şahinaslan ve diğ., 2022) çalışmasında Youtube platformu verilerinin pozitif negatif veya nötr olduğunu tespit eden bir duygu analizi çalışması yapılmıştır. Veriler Youtube API üzerinden elde edilmiştir. 15.082 adet veri üzerinden Weka yazılımı kullanılan çalışmada NB yöntemi kullanılarak sınıflandırma yapılmıştır. 5.167 adet verinin olumlu, 4.644 verinin olumsuz, 5.271 verinin ise nötr olduğu gözlemlenmiştir. Çalışmanın sonucunda doğru olarak sınıflandırılan başarı oranı %65,56 bulunmuştur.

(Tuzcu, 2020) çalışmasında çevrimiçi kitap satış sitesinden 47 adet kitabın okuyucu yorumlarını veri seti olarak kullanmış ve duygu sınıflandırması yapmıştır. 91309 adet yorum bulunan bu veri setinin pozitif negatif olarak sınıflandırma başarı oranları incelenmiştir. Yöntem olarak MLP (Multi-layer perceptron/MLP), NB, DVM ve lojistik regresyon kullanılmıştır. MLP başarı oranı %86 NB başarı oranı %81 DVM başarı oranı %65,71 Lojistik regresyon başarı oranı ise %79 olduğu gözlemlenmiştir.

(Toçoğlu ve diğ., 2019) çalışmasında Türkçe metinlerde duygu analizi yapmayı amaçlamışlardır. 5 bin katılımcı ile bir anket yapılmış ve anket sonucunda 4709 katılımcının dokümanları onaylanmıştır. 27350 belge toplanmıştır. Bu verileri mutluluk, korku, öfke, üzüntü, tikslenme ve şaşırma gibi kategorilere ayırmışlardır. Veri ön işlemede öznitelik aşamalarından sonra TF-IDF ağırlıklandırma modeli kullanılarak metin vektörizasyonu işlemi yapılmıştır Yöntem olarak yapay sinir ağları, destek vektör makineleri, RO ve EYK algoritmaları kullanılmıştır. Değerlendirme metriği doğruluk değeri kullanılmış ve genel doğruluğa bakıldığında her bir kategori işlemi için YSA algoritması diğer algoritmalara göre daha başarılı olduğu gözlemlenmiştir. En yüksek doğruluk %90,58 değeri ile destek vektör algoritması tikslenme kategorisinde olduğu, en düşük doğruluk %56,44 değeri ile k en yakın komşu şaşırma kategorisinde olduğu gözlemlenmiştir.

(Yürütücü ve Demir, 2023) çalışmasında covid-19 tweetlerindeki metinler üzerinde duygu analizi yapmışlardır. Kaggle üzerinde paylaşılmış olan Twitter'da 2020 yılındaki 1,5 aylık covid-19 tweetlerinden elde edilmiş veri seti üzerinde çalışma yapılmıştır. Eğitim kümesinde 41.157 adet veri bulunmaktadır. Verilerin %60'ı eğitim veri seti, %20'si test, %20'si doğrulama veri kümesi olarak bölünmüştür. Veri setinden hashtag, noktalama işaretleri gibi gereksiz ifadeler çıkarılmıştır. Retweet gibi tekrar eden tweetler veri setinden silinmiştir. Tüm harfler küçük harfe dönüştürülmüştür. Ön işlemeden sonra 40.923 verinin olduğu gözlemlenmiş ve bu verilerin 11381 tanesi olumlu, 9889 tanesi olumsuz, 7560 tanesi nötr olduğu tespit edilmiştir. Bu çalışmada NB, BERT, robert ve bertweet modelleri kullanılmıştır. Performans metriği olarak ise F1 puanı, kesinlik ve duyarlılık ölçümleri kullanılmıştır. F1 puanı pozitif veriler için %73, negatif değerler için %74, nötr değerler için ise %49 oranında bulunmuştur. Genel olarak doğruluk oranı %70, kesinlik oranı ise %70 olarak çıkmıştır. BERT modelinde F1 puanı pozitif için %91,

negatif için %89, nötr için ise %80 olarak tespit edilmiştir. Roberta modelinde F1 puanı pozitif için %90, negatif için %90 ve nötr için %80 olarak hesaplanmıştır. Bertweet modelinde pozitif değerler için F1 puanı %91, negatif değerler için %91 ve nötr değerler için %79 şeklinde hesaplanmıştır.

Tez çalışması, incelenen literatür çalışmalarından farklı olarak duygu analizi alanında Jira yazılım süreç aracındaki yorumlar üzerinde yapılmıştır. Makine öğrenmesi sınıflandırma algoritmaları TF-IDF ve count vectorizer vektörizasyonu yöntemiyle kullanılmıştır. TF-IDF yöntemi kullanılarak EYK, XGBoost, RO ve DVM algoritmalarında parametre optimizasyonu yapılmıştır. BERT, BERT+DVM, BERT+RO, BERT+GNB algoritmaları kullanılmıştır.



3. JIRA

Yazılım süreç tasarımı uzun bir süreçtir. Bu süreci yönetmek için bir araç kullanmak bu süreci kolaylaştırır ve zamandan tasarruf etmeyi sağlar. JIRA, Atlassian firması tarafından 2002 yılında piyasaya sürülmüş, yazılım yönetme süreçleri için kullanılan araçlardan biridir (URL-1).

Çevik yazılım sürecini uygulamak için kullanılabilir. Şirketler bu araç sayesinde iş akışlarını rahatlıkla takip edebilirler. Projelerin detaylı planlanmasını ve takip edilmesini sağlar. Jira'daki üyelere göre işler atanıp projenin ilerleyişi sağlanabilir. Ürün gereksinimlerini iterasyonları yönetmek için kullanılabilir. Ekiplerin hangi işi ne zaman yapacağını takibi yapılabilir. Projede hataların neler olduğu ve hangilerinin düzeldiği görülebilir. Takım arkadaşlarının düzen içinde ve uyumlu çalışmasını destekler. Kolay ve rahat kullanımı mevcuttur. Filtreleme özelliği mevcuttur. Kime atanmış, çözümlenmiş mi, kim tarafından açılmış, hangi tarihte açılmış, hangi tarihte çözümlenmiş gibi filtrelemeler yapılabilir. Kullanıcıların yorumları, eklemeleri ve notları herkes tarafından görülebilir. Bu sayede işlerin düzenli ilerlemesi sağlanabilir. Raporlar oluşturulabilir. Bu raporlar sayesinde ekipteki kişilerin performansı değerlendirilip gelecekteki işlerin planlanmasının daha kolay yapılabilmesi sağlanır. Projenin ne kadar sürede yapıldığı gibi zaman da ölçülebilir. Finans, sağlık, telekomünikasyon gibi çeşitli sektörlerde, yazılım firmalarında, e-ticaret şirketlerinde, teknoloji şirketlerinde yaygın olarak kullanılmaktadır.

Jira, 30 günlük ücretsiz olarak kullanıma açık olup abonelik usulü satın alınabilir bir yazılım takip aracıdır. Bu ücretsiz kullanımdan faydalanılarak tez çalışmasında örnek proje takibi yapılmıştır. Şekil 3.1 'de örnek açılmış bir Olay (Incident) bulunmaktadır. Description kısmında müşterinin yaşadığı sorunun ne olduğu yazılmaktadır. Reporter kısmı bu olay kaydını kimin açtığı, assignee kısmı ise bu olay kaydının kime atandığını gösterir. Priority olay kaydının öncelik derecesini gösterir. Bu şekilde gösterilen olay öncelik değeri kritik olduğunu ifade eder. Şekil 3.2'de ise Jira'da bulunan kayıtlar için filtrelemeye bir örnektir. Atanan kişiler mevcut kişiler ve çözümlenmemiş olanları güncelleme sırasına göre filtrelemiş sonuç ise sol tarafta sıralanmıştır.

Müşteri listesinin gelmemesi

Create subtask Link issue ...



Merve Kul raised this request via Jira

[Hide details](#)

[View request in portal](#)

Description

Müşteri listeleme ekranında 10 numaralı müşteri gelmemektedir. Yardımlarınızı rica ederim.

Affected services



[+ Add service](#)

Urgency High

Impact Significant / Large

SLAs

within on

Mar 19 01:00 PM

Time to resolution
within 36h

Details

Assignee



Unassigned

[Assign to me](#)

Reporter



Merve Kul

Request Type



Report a system problem

Priority



Medium

Responders

0 Responders

Şekil 3.1. Jira'da açılmış olay kaydı örneği

Filter for AP

[Filter details](#)

[Apps](#)

[Share](#)

[Export](#)

[Grid](#)

Basic

[JQL](#)

project = AP and reporter= Merve Kul Order BY Rank ASC



[Go back to filter](#)

[Save filter](#)

Rank



maillerime erişemiyorum

AP-1



Müşteri listesinin gelmemesi

AP-2



2 of 2

AP-1

maillerime erişemiyorum

Create subtask

Link issue

...



Merve Kul raised this request via Jira

[Hide details](#)

[View request in portal](#)

Description

Mailim açılmıyor desteklerinizi rica ederim.

Affected services



[+ Add service](#)

Urgency

High

1



Open



SLAs

Nov 18 03:00 PM



Time to first response
within 6h

Nov 22 01:00 PM



Time to resolution
within 36h

Details

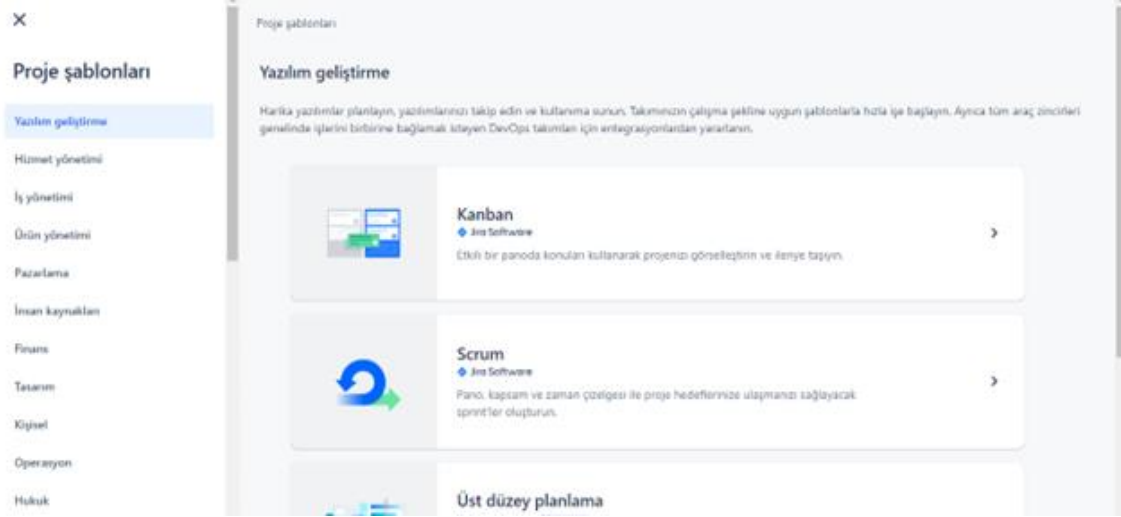
Şekil 3.2. Jira filtreleme örneği

3.1. Jira Software Sürümü

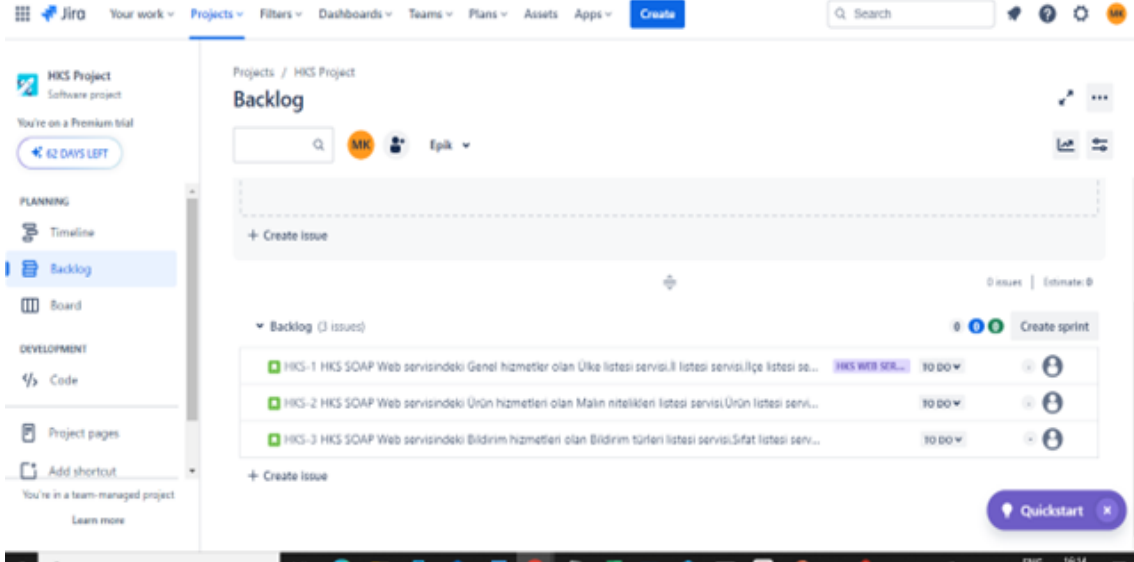
Yazılım geliştirme ekiplerinin projelerini takip etmeleri açısından tasarlanmış bir sürümdür. Çevik yazılım metodolojisini desteklemektedir. Şekil 3.3'deki gibi Scrum ve kanban şeklinde iki çevik model seçilerek ilerlenebilir. Kanban için işlerin görselliğini öne çıkaran bir sistem mevcuttur. Scrum için ise Şekil 3.4'teki gibi iş listesi oluşturulabilir. Kullanıcı hikayeleri oluşturulup, işlerin önceliği belirlenebilir. Şekil 3.5'deki gibi iterasyon planlama yapılabilir. Bu sistemde iterasyon süreleri belirlenebilir, görev atama ve sürecin takibi gibi işlemler yapılabilir. Grafikler ile iterasyonlar izlenebilir. Kişilere görevler atanabilir ve bu atanmış kişiler çalışmaya başladıklarında görevin durumunu “üzerinde çalışılıyor” olarak değiştirebilir. Görev durumları yapılacak, devam eden ve tamamlandı şeklinde takip edilebilir. Projelerin nasıl ilerlediği ve ekibin nasıl performansı olduğu hakkında raporlar oluşturularak takip edilebilir.

Örnek Kullanım Senaryosu:

1. Bir web API yazılım geliştirme projesi başlatılır. Proje türü yazılım geliştirme sürecine uygun olarak scrum veya kanban olarak seçilebilir.
2. Ürün sorumlusu (Product owner) tarafından iş listesi oluşturulur. Örneğin API entegrasyonundaki adımların olduğu kullanıcı hikayeleri oluşturulabilir.
3. İterasyonlar 2-4 haftalık olarak oluşturulabilir ve ekipteki yazılımcıların becerilerine göre bu iterasyonlar atılabilir. Her bir iterasyonda işlerin önceliği belirlenir. Örneğin backend bir geliştirme işlemi yapılacak ise backend geliştiricisine iş ataması yapılır.
4. İterasyonları ilerleten yazılımcılar tamamlanması gereken görevleri yapılacak, iş üzerinde çalışılıyorsa devam ediyor ve işlem yapıldı ise tamamlandı şeklinde durumunu değiştirebilir.
5. Grafiklerden bu iterasyon durumları izlenebilir. Böylece ekip bu grafiklere bakarak işlerin hangi aşamada olduğunu kolayca gözlemleyebilir.
6. Tamamlanan iterasyonlar için birim ve entegrasyon gibi testler yürütülür, değerlendirilir ve sprintin çıktıları göz önünde bulundurularak eksik bir süreç varsa belirlenir. Sonrasında yeni bir iterasyon planlanır. Bu işlem döngüsel olarak bu şekilde devam eder. Her sprint geliştirmeye ve iyileştirmeye açıktır.



Şekil 3.3. Jira proje şablon seçimi



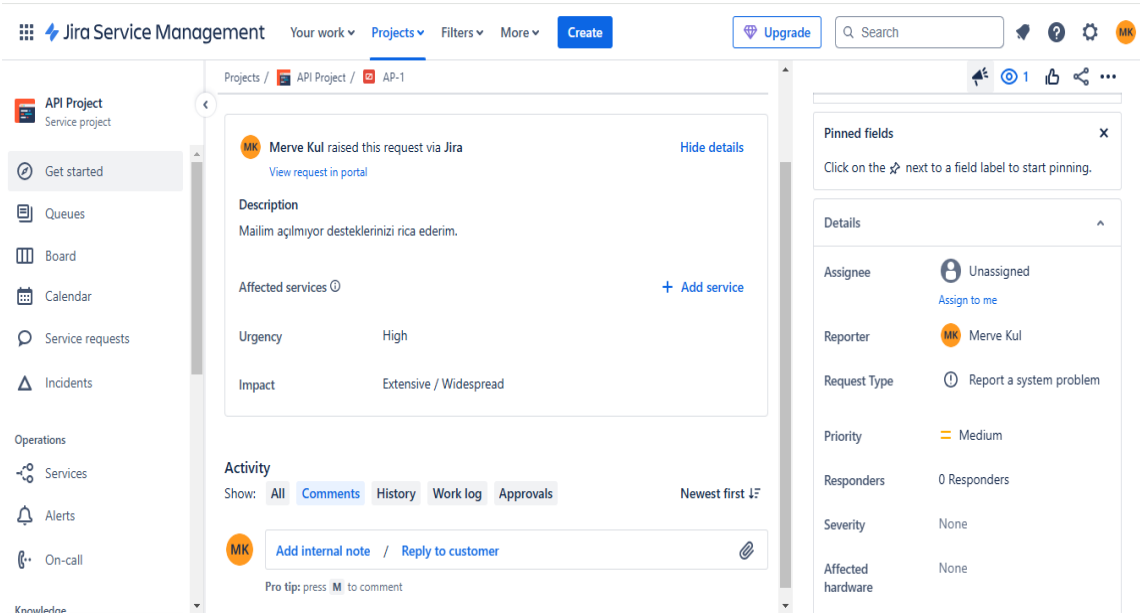
Şekil 3.4. İş listesi Oluşturma Örneği



Şekil 3.5. İterasyon Planlama

3.2. Jira Service Management Sürümü

Bu Jira sürümü, anlık hataların çözümü için bilgi teknolojileri ekiplerinin ve müşteri hizmetleri tarafından kullanılmaktadır. Müşterilerin sistemin nasıl çalıştığını öğrenebilmeleri için Confluence adında Atlassian firması tarafından geliştirilen proje dokümantasyonu bulunmaktadır. Bu sayede müşteriler sorunlarına kendileri de çözüm bulabilirler. Sistem kesintisi gibi anlık sorunlar için Jira üzerinden kayıt açılabilir. Örneğin bir kullanıcının e posta erişiminde bir sorun yaşıyorsa bu sorunun giderilmesi için olay kaydı açılır ve bu kaydın önem seviyesi belirlenir. Bir kişiye atanır ve bu kişi çözdükten sonra kaydı kapatır. Kaydın Hizmet Seviyesi Anlaşmaları (SLA) kapsamında belirlenmiş bir çözüm süresi bulunur. Bu sürelerle ekibin performansına bakılabilir. Kayıtların altına müşteri memnuniyetini bildiren yorumlar yapabilir. Şekil 3.6 'da e-postasına erişemeyen kullanıcının açtığı olay örneği bulunmaktadır. Atama (assignee) kısmında henüz kimseye atanmamış öncelik seviyesi orta derece olan bir kayıt bulunmaktadır. Yorumlar (comments) kısmı çözüm yapıldıktan sonra memnuniyet veya şikayetlerin belirtilebileceği kısımdır.



Şekil 3.6. Jira 'da açılmış bir olay içeriği

4. MATERYAL VE METOT

4.1. Makine Öğrenmesi

Yapay zekanın bir alt dalı olan makine öğrenmesi bilgisayarların verileri kullanarak öğrenmesi sağlayan bir süreçtir. Makine öğrenmesi 1952 yılında Arthur Samuel tarafından o dönemde oynanan dama oyunu üzerine geliştirilmiştir (Akyol ve Özkan, 2023). Oyun her oynandığında daha iyi oynanma yollarını keşfetmeyi amaçlamıştır. Bu oyunun analiz edip öğrenmesiyle makine öğrenmesinin ilk adımı atılmıştır. Teknolojinin ilerlemesiyle birlikte veri miktarında büyük bir artış meydana gelmiştir. Sadece insanlar tarafından üretilen veriler değil akıllı telefonlar, giyilebilir teknolojik aletler gibi sistemler tarafından üretilen veri üretimi de önemli boyutlara ulaşmıştır. Üretilen bu verileri yönetmek insan gücüyle zor bir hal almıştır. Makine öğrenmesiyle bilgisayarların herhangi bir insan müdahalesi olmadan deneyimleyerek geliştirilmektedir. Makine öğrenmesiyle teknoloji açısından yeni bir çağın kapısı açılmıştır. Çeşitli algoritmalar kullanılarak öğrenme yeteneği kazandırılmış sistemler oluşturur. İnsan zekasını taklit ederek veriyi analiz eder. Geçmiş veriler üzerinde çalışıp yeni veriler üzerinde tahminlerde bulunur. Sağlık, finans, pazarlama, ulaşım gibi birçok alanda kullanılır. Makine öğrenmesiyle analiz etme, karar verme ve tahmin yapma işlemleri yapılabilir. Makine öğrenmesiyle insanların verileri analiz ederek iş yapması açısından kolaylık sağlamıştır. Hata oranını en aza indirir, daha doğru ve daha hızlı sonuçlar elde edilmesini sağlar. Makine öğrenmesi üç gruba ayrılır: Denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) (Zilyas ve Yılmaz, 2023).

4.1.1. Makine Öğrenmesi Yöntemleri

Denetimli öğrenme (Supervised learning): Etiketlenmiş veri kümeleri kullanılır. Bir girdi verilerine karşılık olarak hangi çıktı verisinin geldiği belli olan veri setleri ile çalışılır. Bu çıktısı belli olan verilerle öğrenilen algoritma sonucu yeni verilere karşılık gelecek olan çıktı verileri üretilir (Tosunoğlu ve diğ., 2021). Örnek olarak kedi köpek ve kuş fotoğrafları verilip her birinin kedi mi kuş mu köpek mi olduğu çıktı verisi olarak verildiği veri setidir. Bu verilerle çalışan model yeni bir veri geldiğinde hangi kategoriye ait

olduğunu bulacaktır. Etiketli veriler kullanıldığı için yüksek doğruluk oranı sağlayabilir fakat dezavantaj olarak model bu verilere aşırı uyum sağlayabilir. Etiketleme yapmak büyük veri setleri için maliyeti artırabilir. Veri setinin dengesi büyük önem taşır. Eğer dengesiz sınıflandırma yapılmışsa bu durum modelin performansını düşürebilir. Regresyon ve sınıflandırma görevlerinde kullanılır. DVM, YSA, LR, NB, EYK, RO ve KA algoritmaları denetimli öğrenme modeline örnek algoritmalarındandır. (Bilgin, 2017)

Sınıflandırma: Sınıfı belli olmayan verinin hangi sınıfta olduğunu bulan denetimli öğrenme biçimidir. Sınıflandırmanın amacı yeni bir verinin hangi sınıfa ait olduğunu belirlemektir. Veri setinin kategorilerine göre iyi ve kötü, büyük ve küçük, doğru ve yanlış gibi sınıflandırmalar yapılabilir. E-posta spam filtreleme, covid 19 teşhisi, görüntü sınıflandırması, duygu analizi, kredi risk değerlendirmesi gibi alanlarla sıklıkla kullanılmaktadır. Makine öğrenmesinde ikili, çoklu ve çok etiketli sınıflandırma görevleri bulunmaktadır.

İkili sınıflandırmada etiketli verinin iki sınıfının olduğu durumlarda girdi verisinin bu iki sınıftan birine atar. Amaç girdi verilerini iki farklı kategoriden birine sınıflandırmaktır. Örneğin ikili biçimde etiketlenmiş doğru ve yanlış, pozitif ve negatif, hasta ve sağlıklı, araba ve uçak gibi kategorilerden birine sınıflandırma işlemi yapmaktır. Lojistik regresyon ve destek vektör makineleri ikili sınıflandırma için tasarlanmış algoritmalarındandır. Bu gibi sınıflandırma tahminlemelerinde bu algoritmalar tercih edilebilir.

Çok sınıflı sınıflandırma sınıf sayısının ikiden fazla olduğu sınıflandırmadır. Çok sınıflı sınıflandırmada etiketli verinin çıktıları ikiden fazla kategoriye ayrılmıştır. Bu sınıflandırmada amaç ikiden fazla sınıflandırma kategorisinden, girdi verisinin hangi sınıfa ait olduğunu tespit etmektir. Örneğin kamyon, otomobil, traktör kategorileri bulunan bir veri setinde yeni bir girdinin hangi sınıfa ait olduğunun tespitini yapmaktır. K en yakın komşu, rastgele orman gibi algoritmalar tercih edilerek sınıflandırma tespiti yapılabilir (Özkan ve Kar, 2022).

Çoklu etiketli sınıflandırmada etiketli verilerde her veri birden fazla sınıfa ait olabilir. Örneğin bir görüntüde insan, köpek ve uçak olup bu resmin sınıflandırması yapılacaktır.

olabilir veya bir film türü hem romantik hem komedi olabilir. Bu gibi sınıflandırmalar çoklu etiketli sınıflandırmaya örnektir. RO, YSA algoritmaları tercih edilerek sınıflandırma yapılabilir (Erçire ve Ünsal, 2024).

Regresyon: Verinin çıktısı sürekli olan değerleri tahmin etmede görevli bir denetimli öğrenme biçimidir. Regresyonun amacı girdi değerlerine göre çıktı değerlerini tahmin etmektir. Bir evin oda sayısı ve metrekaresi gibi özelliklerine göre fiyatını, çalışma yılı ve kıdem özelliklerine göre bir kişinin maaşını, nem ve basınç gibi özelliklere göre hava durumunu tahmin etmek için kullanılabilir.

Denetimsiz Öğrenme (Unsupervised Learning): Denetimli öğrenme yönteminin aksine etiketlenmemiş veri kümeleri kullanılır. Etiketsiz veriler içindeki örüntüyü bulup bir algoritma kuran makine öğrenmesi yöntemidir. Veriler üzerinde daha önce bilinmeyen veriler ortaya çıkabilir. Eldeki sadece girdi verileriyle sonuç çıkarılmaya çalışılır. Algoritma sonuçları kesinlik vermez. Tamamen bu sonuç doğrudur veya yanlıştır gibi bir sonuç üretmez. Hiperparametre seçimine göre model farklı sonuçlar verebilir. Veri setinin etiketsiz olması modeli gruplamayı zorlaştırır. (Kızılkaya ve Oğuzlar, 2018). Kümeleme ve ilişkilendirme görevlerinde kullanılır.

Kümeleme etiketsiz veriler içerisinde benzerlik oranına göre gruplandırma işlemine kümeleme denir (Aytaç, 2021). Örneğin belirli bir insan grubunun yaşlarının olduğu veri setinde yaşların birbirine yakınlığına göre genç, orta ve yaşlı olarak kümelenebilir. Bir mağazada birlikte satış yapılan ürünlerin tespiti kümeleme ile bulunabilir.

İlişkilendirme veriler arasındaki ilişki çıkarılarak daha sonrasında bir algoritma üretilerek sağlanan makine öğrenmesi yöntemidir. Veriler arasındaki frekanslara göre gruplandırma yapılır. Markette ekmeğin yanında yumurta mı alınır süt mü gibi analizler yapılabilir. Bu analize göre market tezgahlarında düzenleme gerçekleştirilebilir.

Pekiştirmeli Öğrenme (Reinforcement Learning): Etiketlenmiş verilerle değil deneme yöntemiyle öğrenme algoritması kuran bir makine öğrenmesi yöntemidir. İnsanların ve hayvanların öğrenme yöntemine benzetilir. Makinenin belirli bir hedefe ulaşması gerekir. Bu hedefe ulaşırken birçok deneme yolu izler. Her yaptığı iş için iyi ise ödül, kötü ise ceza yöntemi belirlenir. Bu yöntemler takip edilerek bir algoritma kurulmuş olur. Deneme

yanılma yöntemi kurulduğu için tek bir sonucun doğruluğu değil uzun vadeli çalışmalarda daha başarılıdır. Bu nedenle yavaş öğrenir. Karmaşık problemlerin çözümünde çok fazla deneme yapması gerekir. Bu denemeleri yapması için büyük bellek kapasitesine ihtiyaç duyulur. Etiketli veriye ihtiyaç duymaması ise maliyeti azaltmada yardımcı olur. İlaç tedavileri, en iyi rotayı bulma, robot hareketleri gibi alanlarda yaygın olarak kullanılan bir makine öğrenmesi yöntemidir (Köylü ve Atıkan, 2024).

4.1.2. Makine Öğrenmesi Süreci

Problem Belirleme: Bu aşamada makine öğrenmesi ile çözümlenebilecek problem belirlenir. Projenin hedefi, kapsamı, istenen sonuçları ve gereklilikleri bu aşamada netleşir. Makine öğrenimi süreci için sağlam bir temel oluşturmak önemlidir.

Veri Toplama: Belirlenen problemin çözümü için manuel olarak, API'ler veya veri tabanları gibi çeşitli kaynaklardan soruna yönelik veriler toplanır. Bu aşama çözümün kalitesi için de kritik bir aşamadır. Makine öğrenmesinin temel aşaması olan bu aşamada verilerin güncelliği, doğruluğu ve yeterliliği önemlidir. Eksik verilerle çalışan algoritmaların performansı düşük olabilir. Verilerin kalitesi ne kadar iyi olursa modelin tahminlemesi ve performansı aynı oranda artar.

Veri Ön İşleme: Modeli eğitmek ve doğru sonuca ulaşmak için ham verilerle çalışmak uygun olmayabilir. Ham veride eksik, dağınık, gereksiz veriler bulunmaktadır. Bu da modeli eğitmekte ve doğru sonuca ulaşmakta yanlış yönlendirmektedir. Veri ön işlemenin amacı veriyi makine öğrenmesi modeline uygun hale getirmektir (Polatgil, 2023).

Veride eksik değerler var ise pandas kütüphanesinin isnull fonksiyonu ile bulunur (URL-2). Bu değerler çok sayıda değilse, veri setinden çıkarıldığında model üzerindeki etki bozulmayacaksa veri setinden pandas kütüphanesinin dropna fonksiyonu ile çıkarılabilir. Eğer çıkarılmak istenmiyorsa veya çıkarıldığında modele etkisi büyük olacaksa sayısal değerler için ortalama, medyan, mod gibi tekniklerle pandas kütüphanesinin fillna fonksiyonu ile doldurulabilir. Kategorik verilerde ise en sık olan kategoriye göre değer ile doldurulabilir. Veri setinde gereksiz modele etki etmeyecek veriler var ise çıkarılabilir. Örneğin Id gibi bir değer var ve bu modelin eğitilmesinde bir faydası olamayacak ise Id sütunu veri setinden çıkarılabilir.

Veri setinde yinelenen kayıtlar var ise pandas kütüphanesi duplicated fonksiyonu kullanılarak bulunabilir. Bu yinelenmiş veriler gereksiz ve farklı olarak ek bilgi içermiyorsa pandas kütüphanesinin drop_duplicates fonksiyonu ile veri setinden çıkarılabilir.

Makine öğrenmesi algoritmaları sayısal verilerle çalıştığı için veri setinde kategorik veriler var ise sayısal değerlere dönüştürülür. Scikit-Learn kütüphanesinin LabelEncoder, OneHotEncoder fonksiyonlarıyla dönüştürme işlemleri yapılabilir (URL-3). Örneğin evet ve hayır olmak üzere iki kategori bulunuyorsa bu veriler 0 ve 1 şeklinde sayısal verilere dönüştürülebilir (Kasongo Dahouda ve Joe, 2021).

Veri normalizasyonu yapılarak veriler 0 ile 1 veya -1 ile 1 aralığında değerlere dönüştürülebilir. Normalizasyonun amacı değerlerin belirli bir aralığa getirilip modelin daha iyi ve hızlı performans vermesini sağlamaktır. Normalizasyon yapılırken değer aralığındaki farklılığın bozulmaması önemlidir. Veri setinde çok yüksek veya çok düşük sayılı veriler olabilir. Bu yöntem ile veriler benzer bir aralığa getirilerek modelin daha hızlı ve kolay eğitilmesi için veri seti hazır hale getirilmiş olur.

Model Seçimi ve Eğitimi: Belirlenen probleme uygun olan model seçimi yapılır. Doğru modelin seçilmesi modelin performansı açısından önemli rol oynar. Sınıflandırma yapılacaksa sınıflandırma algoritmalarından en uygunu, kümeleme yapılacaksa kümeleme algoritmalarından en uygunu seçilir. Veriler eğitim ve test veri seti olarak ayrılır. Eğitim veri setiyle seçilen algoritma öğrenir ve test veri seti üzerinde tahminleme yapılır. Seçilen algoritmanın hiperparametre seçimi de önemlidir ve modelin performansını etkiler.

Model Değerlendirmesi ve Optimizasyonu: Test seti üzerinde model eğitildikten sonra değerlendirme işlemi yapılır. Regresyon algoritmaları kullanıldı ise ortalama mutlak hata veya ortalama kare hata değerlendirme metrikleri kullanılabilir. Sınıflandırma algoritmaları kullanıldı ise doğruluk, kesinlik, duyarlılık veya F1 puanı gibi değerlendirme metrikleri kullanılabilir. Model değerlendirmesi beklenen performansın altında kalırsa doğruluğunu ve performansını artırmak için hiperparametre optimizasyonu

yapılabilir. Veri eksikliği varsa veri eklenebilir. Birden fazla model birleştirilerek yeni bir modelle eğitilip test edilip değerlendirme işlemi yapılabilir.

Model Dağıtımı: Model eğitildi ve sorunsuz bir şekilde çözümlendi ise gerçek hayatta kullanıma sunulabilir. Bu çözüm bir web sitesine entegre edilecek ise sisteme yüklenir. API uygulaması şeklinde kullanılabilir.

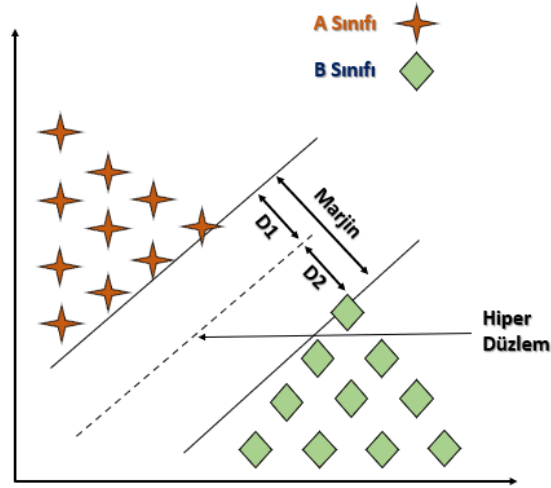
Model İzleme: Model canlı hayatta kullanıma sunulduktan sonra performansı izlenir. Gerçek verilerle çalıştığında performansı düşerse geliştirme güncelleme işlemleri yapılır. Her türlü yeniliğe açık olmalıdır.

4.1.3. Sınıflandırma Algoritmaları

Destek Vektör Makinesi (DVM): DVM'ler, veri sınıflandırma ve regresyon analizinde kullanılan güçlü bir tekniktir. Veri setindeki örneklerin hangi sınıflara ait olduğunu öğrenmeye çalışarak daha önce görülmemiş olan verilerin sınıflarını tahmin eder.

Destek vektör algoritması elimizde bulunan sınıflandırma verilerine en uzak mesafede bir çizgi çizer. Buna hiper düzlem denir. Hiperdüzlem gruplandırılacak verilere bakılarak en yakın iki verinin ortasında olan bir çizgidir. Bu çizgiye göre sınıflandırma yapılır. Veri ve hiperdüzlem arasındaki boşluğa ise marjin denir. Marjinin büyük olması sınıflandırılacak veriler arasındaki boşluğun fazla olması demektir ve bu sınıflandırmanın daha doğru yapılabileceği anlamına gelir. Şekil 4.1'de A ve B sınıfının destek vektör makinesi algoritması ile nasıl sınıflandırıldığı gösterilmektedir. Birbirine en yakın A ve B sınıfındaki verilere birer destek vektör çizilmiş ve sınıflar arası en iyi boşluk(marjin) bulunarak bir hiperdüzlem oluşturulmuştur. Bu algoritma modellenirken c parametresi alır. C küçük ise marjin dar anlamına gelmektedir.

DVM'ler büyük veri setleriyle çalışır fakat yüksek maliyet gerekir. Orta ve küçük veri setlerinde yüksek doğruluğa ulaşır. En sık kullanılan algoritmalarından biridir. DVM'lerde hiperparametrelerin doğru seçimi performansı etkiler. Dengesiz veri setleri modelin performansı düşebilir. İkili sınıflandırmalarda daha başarılı olsalar da çoklu sınıflandırma işlemleri de yapılabilir. Hisse senedi tahmini, e-posta filtreleme, el ve yüz tanıma gibi alanlarda kullanılır.



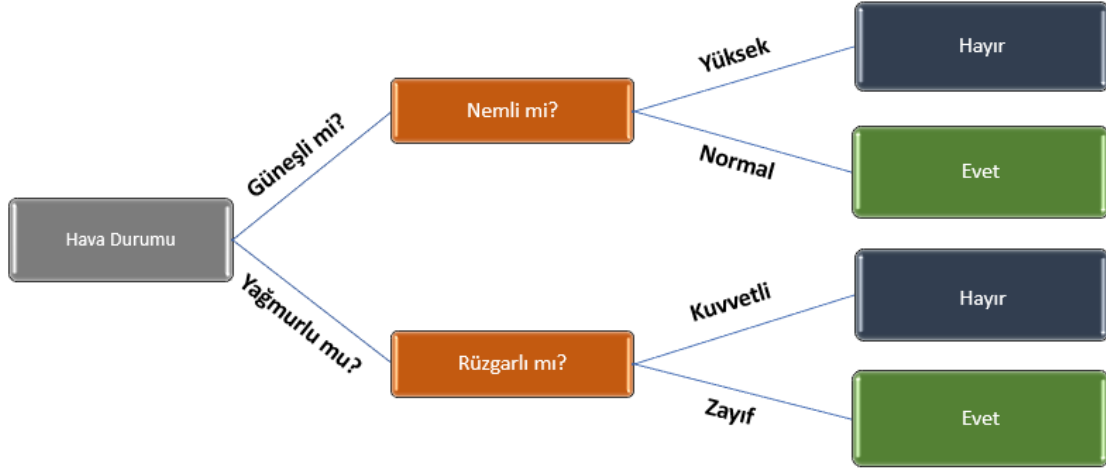
Şekil 4.1. DVM modeli çalışma yapısı

Karar Ağacı (KA): Karar ağacı algoritması basit bir karar ağacı modelidir. En çok kullanılan denetimli öğrenme algoritmalarından biridir.

Sınıflandırma ve regresyon için kullanılabilir. Bir ağaca benzer yapısı vardır. Tüm veri setinin yer aldığı ilk kısım kök, dalların uzanan uç kısımları düğüm ve en sonunda ise belli bir kurala göre karar verilen kısım olan yapraklar bulunur. Bir önceki eğitim verilerinden öğrenilen bilgiyle bir sonrakini tahmin eder. Şekil 4.2 'de karar ağacı çalışma yapısını ifade eden bir örnek yer almaktadır. Her satır bir kararı belirtir ve sonraki her satır ise bu kararın sonuçlarından birini temsil eder. Hava durumuna göre dışarı çıkılabileceğinin tespiti için öncelikle havanın güneşli mi yoksa yağmurlu mu olduğuna bakılır. Eğer hava güneşli ise nemli olup olmadığına bakılır. Yağmurlu ise rüzgarlı olup olmadığına bakılır. Bu şekilde devam eden ağacın sonucuna bakılarak dışarı çıkıp çıkmama durumuna karar verilir. Ağacın ilk hücresi kökü, altındakiler düğümleri en alttakiler ise yaprakları ifade eder. Karar ağaçlarında kurallar açık bir şekilde belli olduğu için anlaşılması kolaydır. Karar ağaçları çok derinleştğinde eğitim setinde aşırı uyum problemi ortaya çıkabilir.

Parametrelerinden kriter fonksiyonlarından yaygın olarak gini ve entropy kullanılır. Kriter fonksiyonları ağaçların düğümlerinin bölünmesinin nasıl olacağını belirlemek amacıyla kullanılır. Max_depth ağacın derinliğidir ve ağacın karmaşıklığını veya

basitliğini belirler. Min_samples_leaf en az yaprak sayısı anlamına gelir ve en az yaprağa ulaşılan kadar ağaç dallamaya devam eder.



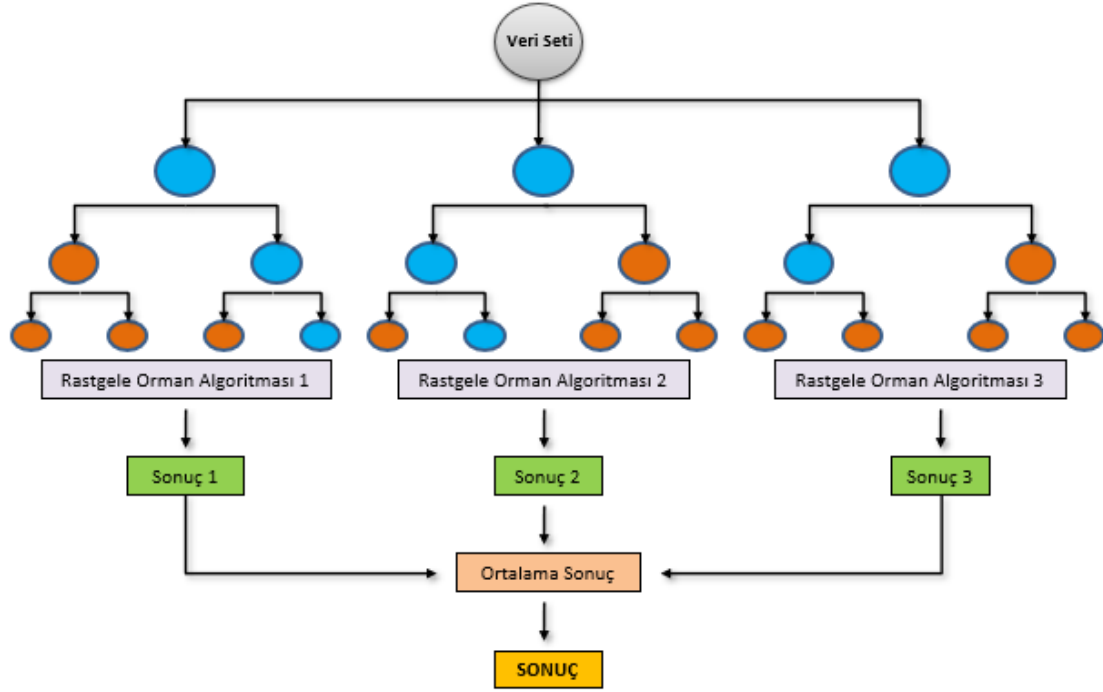
Şekil 4.2. KA modeli örnek senaryosu

Rastgele Orman (RO): Sınıflandırma ve regresyon problemlerinde sıkça kullanılan algoritmalarından birisidir.

Eldeki verilerden rastgele olarak veriler çekilip her biri için karar ağacı oluşturulur. Birden fazla karar ağacı verisinin bir araya gelmesiyle oluşur. Bu verilere bir karar ağacı modeli kurulur. Her bir alt veriden bir karar çıkar ve en iyi karar sonucu seçilmesi işlemi ile karar verilir. Birden fazla ağaç kullanıldığı için yüksek doğruluk verir. Müşteri analizi finansal analiz gibi birçok yerde kullanılır. Çeşitli karar ağaçlarının bir araya gelmesinden kaynaklı eğitim verilerinin ezberlemesini önlemiş olur. Çeşitli ağaçlar oluşturup iyi bir performansa ulaşmak için büyük veriye ihtiyaç duyar. Şekil 4.3'te rastgele orman algoritmasının çalışma yapısı gösterilmiştir.

Rastgele orman algoritmasında kullanılan ağaç sayısı, özellik sayısı, ağaçların derinliği dallanma için minimum örnek sayısı, yaprak düğümleri için minimum yaprak sayısı gibi önemli parametreler kullanılmaktadır. Ağaçların sayısı arttıkça aynı oranda modelde karar ağacı bulunacaktır. Karar ağaçlarında aşırı öğrenme olabilir fakat rastgele orman algoritmasında birden fazla ağaç olduğundan dolayı bu riski azaltmış olacaktır. Tüm ağaçların tahminleri birleştiğinde her ağacın farklı tahmin yapması daha güçlü bir model oluşmasını sağlar. Birden fazla ağacın birleşmesiyle bellek ve işlem gücü açısından

hesaplama maliyeti artarken genelleme yapımı olduğundan dolayı daha doğru sonuçlar elde edilecektir. Ağaç derinliği ne kadar büyük olursa model o kadar karmaşık olur.



Şekil 4.3. RO modeli çalışma yapısı

Naive Bayes (NB): Olasılığa dayalı bir makine öğrenmesi modelidir. Veri setinde bulunan her bir elemanın olasılığını hesaplar ve olasılığı en yüksek olan değere göre sınıflandırma işlemi yapılır. Denklem (4.1)'de bayes teorem denklemi gösterilmiştir. B doğru kabul edildiğinde A'nın doğru olma olasılığı hesaplanır (İlhan ve Sağaltıcı, 2020). Spam filtreleme, duygu analizi gibi uygulamalarda başarılı bir şekilde kullanılmaktadır. Büyük veri setlerinde yüksek performansa sahiptir. Uygulaması basit olan gözetimli öğrenme algoritmalarındandır. Uygulaması kolay ve düşük maliyetlidir. Dezavantajı olarak daha önce görülmemiş olan durumlara sıfır olasılık olarak atama olursa performans açısından zayıf bir genelleme yapılmış olur.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4.1)$$

$P(A|B)$: B doğru olduğunda A'nın olma olasılığı

$P(B|A)$: A doğru olduğunda B'nin olma olasılığı

$P(A)$: A'nın olma olasılığı

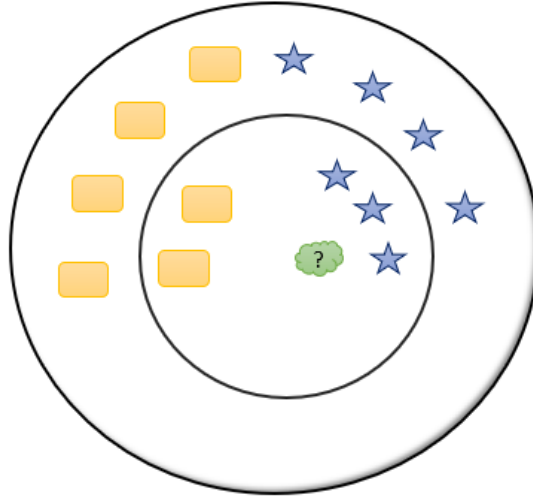
$P(B)$: B'nin olma olasılığı

XGBoost: Sınıflandırma ve regresyon problemlerinde kullanılan bir algoritmadır. Extreme Gradient Boosting anlamına gelmektedir. Gradient Boosting algoritmasının performansının arttırılmış halidir. Gradient Boosting zayıf öğrencilerin bir araya gelerek güçlü öğrenme tekniğini oluşturmaktadır. Zayıf öğrenci tek başına çalıştığında tahminleme yeteneği düşük olan modeldir. Güçlü öğrenci ise tek başına çalıştığında yüksek tahmin yeteneğine sahip olan algoritmadır. Karar ağaçları karmaşık ilişkilerde modelleme yetenekleri düşük olduğundan dolayı genellikle zayıf öğrenci olarak tanımlanabilmektedir. XGBoost algoritması önemli bir ön işlem gerektirmeden eksik değerler içeren verileri de işleyebilir. Aynı anda birden fazla hesaplama yaparak paralel hesaplama tekniği ile büyük veri setlerinde hızlı çalışır (Yaman ve Tekin, 2023).

Öğrenme oranı, maksimum derinlik (ağacın derinliğini temsil eder) ve kullanılacak ağaç sayısı ($n_estimators$) XGBoost algoritmasının performansını etkileyen önemli parametrelerdendir. Veri setine göre bu parametreler değişebilir.

En Yakın Komşu (EYK): Basit aynı zamanda etkili sınıflandırma ve regresyon algoritmalarında kullanılan bir algoritmadır.

Bir k değeri belirlenir. Sınıflandırılacak verinin en yakın komşu sayısı bu k değerine göre seçilir. K değeri algoritmanın başarısını doğrudan etkilediği için k değerini iyi seçmek gerekir. Örneğin gridsearch kullanılarak hiperparametre optimizasyonu ile k değeri aralığı verilerek en iyi parametre bulunabilir veya basit bir yöntem olarak algoritmayı birçok k değeri ile çalıştırıp en başarılı değeri almak olabilir. K tane seçilen en yakın komşulara bakılarak veri sınıflandırılır (Demir ve diğ., 2018). Yeni olan bir veri k tane komşularına bakılıp bunların içinden en yakın komşusunun sınıfına dahil olur. Şekil 4.4'te EYK modeli çalışma yapısı gösterilmiştir. Şekilde ifade edildiği gibi üçgen $k=2$ olursa kare grubuna dahil olur fakat $k=5$ seçilirse yıldız grubuna dahil olur. Mesafeler öklid, manhattan, minkowski, hamming gibi birçok hesaplama yöntemi kullanılarak hesaplanır. Uygulanması kolay bir algoritmadır.



Şekil 4.4. EYK modeli çalışma yapısı

4.1.4. Mesafe Ölçüm Metrikleri

Öklidyen mesafe: İki nokta arasındaki mesafe ölçüm metriğidir. X ve y noktalarının farklarının kareleri toplamı ile hesaplama yapılır. Denklem (4.2)'de öklidyen mesafe formülü verilmiştir. Bu denklem için x_1, y_1 ilk noktanın koordinatları x_2, y_2 ikinci noktanın koordinatlarıdır. X noktaları arasındaki farkın kareleri ve y noktaları arasındaki farkın kareleri toplamının kökü alınarak hesaplama yapılır. En sık kullanılan mesafe metriklerinden biridir.

$$d(P_1, P_2) = \sqrt{\{(x_2 - x_1)^2 + (y_2 - y_1)^2\}} \quad (4.2)$$

$P_1 (1,2)$ ve $P_2 (4,6)$ olduğu durumda hesaplaması Denklem (4.3)'te gösterildiği şekilde yapılmaktadır.

$$d(P_1, P_2) = \sqrt{\{(4 - 1)^2 + (6 - 2)^2\}} = \sqrt{9 + 16} = \sqrt{25} = 5 \quad (4.3)$$

Manhattan mesafe: İki nokta vektörlerinin mutlak mesafeleri toplamıdır. Denklem (4.4)'te de manhattan mesafe formülü verilmiştir. Bu denkleme göre x_1, y_1 birinci nokta x_2, y_2 ikinci nokta ise x noktalarının mutlak farkı ile y noktalarının mutlak farkları manhattan mesafe hesaplamasını yapılır.

$$d(P_1, P_2) = |x_2 - x_1| + |y_2 - y_1| \quad (4.4)$$

$P_1 (1,2)$ ve $P_2 (4,6)$ olduğu durumda hesaplaması Denklem (4.5) 'te gösterildiği şekilde yapılır.

$$d(P_1, P_2) = |4 - 1| + |6 - 2| = 3 + 4 = 7 \quad (4.5)$$

Minkowski mesafe: Öklid ve manhattan mesafe formüllerinin genişletilmiş halidir. Parametre p 'nin 1 olduğu durumlarda manhattan formülüne 2 olduğu durumlarda ise Öklid formülüne dönüşmektedir. Öklid ve manhattan a ihtiyaç duyulduğunda geçiş yapması nedeniyle kullanışlıdır. Minkowski formülüne bu durum esneklik kazandırır ve birçok veri analizinde kullanılmaktadır. p arttıkça vektörler arası büyük farklara odaklanılacağından dolayı küçük farklar ihmal edilir. Farklar arasındaki önemli olan değerler bu durumda göz ardı edilebilir. Örneğin yüz tanıma gibi işlemlerde kullanıldığında pikseller arası mesafe ölçülür. Bu gibi durumlarda küçük farkların göz ardı edilmemesi gerekmektedir Parametre bu nedenle doğru seçilmelidir. Denklem (4.6)'da minkowski mesafe formülü gösterilmektedir.

$$d(P_1, P_2) = (\sum_{i=1}^n (|x_i - y_i|)^p)^{1/p} \quad (4.6)$$

$P_1(1,2)$ ve $P_2(4,6)$ ve $p=3$ olduğu durumda hesaplaması Denklem (4.7)'de gösterildiği şekilde yapılmaktadır.

$$d(P_1, P_2) = (|4 - 1|^3 + |6 - 2|^3)^{1/3} = (3^3 + 4^3)^{1/3} = (27 + 64)^{1/3} = (91)^{1/3} \approx 4.5 \quad (4.7)$$

Hamming mesafe: İki eşit uzunluktaki bit dizisi arasındaki mesafeyi ölçer. Tablo 4.1'de hamming mesafe hesaplaması verilmiştir. İki bir karşılaştırılmak istendiğinde uzunluklarının aynı olup olmadığı kontrol edilir. Aynı ise karşılaştırma yapılır ve bitler farklı ise 1 aynı ise 0 olarak döner. Görüntü analizi veya metin analizindeki karşılaştırmalar için, verilerin iletilip iletilmediğinin kontrolü için sıklıkla kullanılır.

Girdilerin 11010 ve 10011 olduğu durumlarda;

Karşılaştırma:

Tablo 4.1. Hamming mesafe hesaplama

Birinci Girdi	İkinci Girdi	Sonuç
1	1	0 (Aynı)
1	0	1 (Farklı)
0	0	0 (Aynı)
1	1	0 (Aynı)
0	1	1 (Farklı)

Mesafe Sonucu: $0+1+0+0+1=2$

4.2. Doğal Dil İşleme

Doğal dil işleme yapay zekanın bir alt dalıdır. Doğal dil işleme bilgisayar ile insan arasındaki dildeki etkileşimi sağlar. Dili anlar ve bilgisayarın anlayacağı şekilde yorumlama yapar. E-postalar, sosyal medya, Alışveriş sitelerinin yorumları, çağrı merkezlerinin kayıtları gibi birçok yerde duygular yazılır. Hukuk, sağlık, finans, ticaret gibi birçok sektörde bu kayıtlar manuel incelenmek yerine doğal dil işleme ile duygular ön plana çıkarılabilir. Kategorize etme işlemi yapılabilir. Örneğin bir metnin sanat, sağlık veya hukuk gibi hangi başlık altında kategorize edilebileceği bulunabilir. Bir metin İngilizce'den Türkçe'ye çevrilirken insan müdahalesi olmadan doğal dil işleme ile yapılabilir. Doğal dil işleme aynı zamanda sohbet robotları ve sanal asistanlar gibi uygulamalarda da kullanılmaktadır. Doğal dil işlemenin tarihi 1950'lere dayanmaktadır. 1980'li yıllarda gramer kuralları üzerine işlenmiştir. 2000'lerden günümüze kadar olan süreçte makine öğrenmesinin denetimli ve denetimsiz öğrenme algoritmaları kullanılarak sohbet robotları sanal asistanlar gibi birçok uygulamalar geliştirildi.

4.2.1. Doğal dil işleme ön hazırlık adımları

Tokenizasyon: Metnin noktalama işaretleri veya boşluklara göre parçalara ayırma işlemine tokenizasyon denir. Bir metni daha küçük parçalara ayırma yoludur. Bu işlem her bir kelimeyi ayrı ayrı ele alacağı için yapılacak işlemi kolaylaştırır. Bilgisayarın metindeki geçen cümleyi anlaması için öncelikle kelimeleri anlaması gerekmektedir. Bu nedenle metni küçük parçalara ayırmak en öncelik olmalıdır. Tokenizasyon işlemi ile kelimeleri ayırıp bu verilerle çalışma yapmak işlemi kolaylaşır hale getirir.

Kelime Tokenizasyonu: Bir metni kelime kelime parçalara ayırır. Bunu metindeki boşluklara veya noktalama işaretlerine göre yapabilir. Örneğin;

Cümle: “Bugün işe gideceğim.”

Token: [“Bugün”, ”işe”, ”gideceğim”, ”.”]

Alt Kelime Tokenizasyonu: Kelimeleri daha küçük birimlere ayırır. Zengin dillerde veya nadir olarak geçen sözcüklerde işe yarayabilir.

Kelime: “Kitaplık”

Token: [“kitap”, ”lık”]

Karakter Tokenizasyonu: Metinde geçen kelimeleri tek tek karakterlere böler.

Cümle: ”Geliyorum.”

Token: [“G”, ”i”, ”l”, ”i”, ”y”, ”o”, ”r”, ”u”, ”m”, ”.”]

Cümle Tokenizasyonu: Metni cümlelere böler.

Metin: “Bugün erken kalkacağım. İşe gideceğim.”

Token: [“Bugün erken kalkacağım.”, ” İşe gideceğim.”]

N-gram Tokenizasyonu: Metinde geçen art arda gelen n kelimeyi gruplar ve bu gruplar arasındaki ilişkiler yakalanır.

Kelime: ”Günaydın”

Token: [”Gü”, ”ün”, ”na”, ”ay”, ”yd”, ”dı”, ”ın”]

Gereksiz kelimeleri çıkarma: Cümle içerisinde kullanılıp fakat cümlelerin anlamına bir etkisi olmayacak kelimeleri çıkarma işlemidir. Metinlerin içerisinde cümle anlamına etkisi olmayan birçok kelime bulunur ve bu kelimeler çıkarıldığında daha az bellek kullanımını sağlar, model önemli olan kelimeler üzerine yoğunlaşır ve modelin performansının artması sağlanır, daha hızlı sonuç verir. Örneğin İngilizce’de “the”

kelimesi cümle anlamını değiştirmez. Türkçe’de “bu”, ”şu” kelimeleri cümleden çıkarıldığında cümlenin anlamına çok etkisi olmaz. Bu nedenle bu kelimeler cümleden çıkarılabilir ve cümle daha sade bir hale getirilebilir. Python’da nltk kütüphanesinde bulunan stopwords komutu kullanılarak gereksiz kelimeleri cümleden çıkarma işlemi yapılabilir.

Kelime gövdeleme ve kök bulma: Metinlerdeki kelime eklerinin kaldırılmasıyla, kelimelerin köklerinin bulunması işlemidir. Kök bulmada örneğin “saw” kelimesi “see” kelimesinin geçmiş zaman halidir. “Saw” kelimesi için kök bulma işlemi uygulandığında “see” kelimesini bulur. Kelime gövdeleme işlemi uygulandığında ise going kelimesi ele alındığında -ing eki kaldırılarak go şeklinde gövdeleme yapılmış olur. Kök bulma hızlı ve basittir. Hızlı olduğu için hata olasılığı yüksektir. Dilbilgisi kurallarını göz ardı eder. Gövdeleme kök bulma işlemine göre daha yavaştır. Dilbilgisel anlam önemlidir. Kök bulmaya göre daha doğru sonuçlar verir. Gövdeleme için Python’da nltk kütüphanesinde bulunan wordnet kütüphanesi kullanılabilir. Türkçe metinler için ise zemberek kütüphanesi kullanılabilir. Şekil 4.5’te Wordnet kütüphanesi kullanılarak “running” kelimesinin gövdeleme işlemi örnek kodu bulunmaktadır. Sonuç olarak “running” fiiline gövdeleme işlemi uygulandığında “run” kelimesini vermektedir.

```
6
7
8 import nltk
9 nltk.download('wordnet')
10
11 from nltk.stem import WordNetLemmatizer
12
13 lemmatizer = WordNetLemmatizer()
14
15 # Kelimeleri lemmatize etme
16 print(lemmatizer.lemmatize("running", pos="v"))
17
```

Şekil 4.5. Gövdeleme işlemi örnek kod parçası

Kelime Gömme (Word Embedding): Kelimelerin vektörlerle temsil edilme işlemidir. Eş anlamlı kelimeler veya farklı eklerle sahip olup aynı anlamı taşıyan kelimeler arasındaki benzerlikler elde edilebilir. Makine öğrenmesi algoritmaları sayısal ifadeler ile

alışmasından kaynaklı olarak metinsel verilerin sayısal ifadelere dnstrlme iřlemidir (Onan, 2020).

4.2.2. Duygu Analizi

Duygu analizi, insanların metin yoluyla anlattığı duygularını sınıflandırmaktır. Doğal dil iřleme ve makine ğrenimi teknikleri kullanılarak insanların duygularının olumlu, olumsuz ve ntr olarak sınıflandırmayı amalar.

Duygu analizi, sosyal medyanın bymesiyle daha ok gndeme gelmiřtir. Duygu analizi yaparken “iyi, gzel, mkemmel” gibi szckler olumlu bir ifade tařır. “Kt, iğren” gibi szckler ise olumsuz ifade tařır. Bazı szcklerin olumlu veya olumsuz olduėuna dair kesinlik olmayabilir. Bu gibi durumlarda duygu analizinde zorluklar yařanmaktadır. rneėin soru cmlerinde bu duruma sıklıa rastlanır. “Neresi daha iyi?” cmlesinde iyi szcė yer alsa da olumlu bir ifade olduėu anlamına gelmemektedir. “Bilgisayarım ok hızlı alıřıyor.” Cmlesi olumlu bir ifade tařmasına raėmen hibir duygu barındırmamaktadır. Duygu analizi birok alanda kullanılmaktadır. Bu kullanım ile insanların iř gcn azaltmıřtır. Sosyal medyada insanların yorumlarını analiz etmede, e-ticaret sitelerinde mřterilerin yorumları iin olumlu veya olumsuz olduėu durumun anlařılmasında, finansal yatırımlar iin hisse senedi gibi yorumları analiz etmede duygu durumları n planda tutularak analiz edilir. Bylelikle bu alanda alıřan řirketlerin de daha iyi rn ve hizmet vermesi saėlanabilir (Eyipınar ve diė., 2021).

Szlk tabanlı yaklařım ynteminde duygusal ierik tařıyan kelimelerin ve her bir kelime iin puanının olduėu liste kullanılarak duygu analizi yapılır. rneėin “iyi” kelimesi olumlu +1 puana sahipken “kt” kelimesi olumsuz -1 veya “ben” kelimesi ntr 0 řeklinde puanlama yapılarak duygu puanları ile kelimeler eřleřir. Metindeki her bir kelime bu duygu szlė listesindeki kelimelerle eřleřerek puanlama sonlanır. Toplanan puanlar pozitif ise metnin olumlu olduėu, negatif ise olumsuz, sıfır ise ntr olduėu anlamına gelir. Szlk tabanlı yaklařım uygulaması kolay bir yntemdir fakat alaycı cmlelerde doėru deėerlendirme yapılamayabilir.

Makine ğrenimi tabanlı yaklařım ynteminde duygu analizi yapılacak veriler toplanır ve bu verilerin sınıflandırılması iin etiketli veriler kullanılır. Her bir metin olumlu,

olumsuz ve nötr olarak duygu etiketlemesi ile ilişkilendirilir. Metinde geçen kelimelerin eğer o metni olumlu veya olumsuz olarak etkilemeyecekse metinden çıkartılır. Sözel ifadeler sayısal ifadelere çevrilir. Makine öğrenimi algoritmalarından hangi modelleme kullanılacaksa seçilir. Model verilerin bir kısmı ile eğitilir. Daha sonra geriye kalan yeni verilerle tahminde bulunur ve hangi duygu sınıfına dahil olduğu belirlenir. Algoritmanın iyi çalışabilmesi için büyük veri seti olması avantajdır. Test verisi algoritmaya çok uyum sağlarsa aşırı öğrenme gibi bir durum ile de karşılaşılabilir. Aşırı öğrenmeyi önlemek için veri setini genişletmek, doğru model seçimi yapmak, çapraz doğrulama ile veri setini farklı alt kümelerle ayırarak işlem yapma gibi yöntemler uygulanabilir (Gürsoy ve diğ., 2018).

4.2.3. Metin Vektörizasyonu

Makine öğrenmesi algoritmaları sayısal ifadeler ile çalışmaktadır. Veri setindeki metin alanında sözel ifadeler yer almaktadır. Bu şekilde algoritma verileri işleyemez ve bir sonuç döndüremez. Bu nedenle sözel ifadeleri sayısal ifadelere dönüştürme işlemi yapılmalıdır. Bu çalışmada 2 çeşit sayısal ifadeye dönüştürme yöntemi kullanılmıştır. Bunlar count vectorizer ve TF-IDF (Term Frequency-Inverse Document Frequency) yöntemleridir.

TF, bir kelimenin metin içindeki toplam kelimelere kıyasla ne kadar sık geçtiğini ölçer. En çok geçen kelimeler metindeki en önemli kelimeler olarak kabul edilir (Sjari ve diğ., 2019). Denklem (4.8) 'de TF hesaplama formülü verilmiştir.

$$TF = \frac{\text{Belgedeki toplam terim sayısı}}{\text{Terimin belgede kaç kez geçtiği}} \quad (4.8)$$

IDF, bir kelime bir metinde çok fazla geçiyorsa çok fazla anlam taşımadığını ifade eder. Örneğin İngilizce'de "the", "is" gibi Türkçe'de "ve", "bu", "de" gibi kelimeler olabilir. IDF'in amacı bu kelimeler çok geçtiği için frekansını düşük vermek ve diğer anlamlı kelimelerin frekansını artırmaktır. Frekansı yüksek kelimeler sınıflandırma yapmak için daha değerli olacaktır. Denklem (4.9) 'da IDF hesaplama formülü verilmiştir.

$$IDF = 1 + \log_e \frac{\text{Toplam doküman sayısı} + 1}{\text{İçinde } t \text{ terimi olan doküman sayısı} + 1} \quad (4.9)$$

TF-IDF değeri ise TF ve IDF hesaplamalarından bulunan değerlerin çarpımları sonucu elde edilir. Denklem (4.10) 'da TF-IDF hesaplama formülü verilmiştir.

$$TFIDF = TF \times IDF \quad (4.10)$$

Tablo 4.2'de cümlelerdeki her bir kelime çıkarılır ve kelimelerin frekanslarının ölçümü yapılır.

Tablo 4.2. Kelime frekansı çıkarımı

Cümle	Ayşe	Eve	Geldi	Koşarak
Ayşe eve geldi	1	1	1	0
Ayşe eve koşarak geldi	1	1	1	1

Tablo 4.3'te her bir kelimenin cümledeki oranları hesaplanır. Terim sayısının bir kelimenin metinde kaç kere geçtiğine oranıdır.

Tablo 4.3. TF değeri hesabı

Cümle	Ayşe	Eve	Geldi	Koşarak	Terim Sayısı
Ayşe eve geldi	0.33	0.33	0.33	0	3
Ayşe eve koşarak geldi	0.25	0.25	0.25	0.25	4

Tablo 4.4 'te IDF formülü kullanılarak IDF değeri hesaplanır. Metinlerde sık geçen kelimelerin IDF değeri düşük, nadir geçenlerin ise yüksek olur. Tablo 4.5' te TF ve IDF değerlerinin çarpım sonucunu vermektedir. Bu hesaplama sonucunda dokümanda her kelimenin önemi sayısal ifadelerle belirlenmiştir. Makine öğrenmesi algoritmaları için uygun formata getirilmiştir.

Tablo 4.4. IDF değeri hesabı

Cümle	Ayşe	Eve	Geldi	Koşarak
Ayşe eve geldi	2.1	2.1	2.1	2.38
Ayşe eve koşarak geldi	2.38	2.38	2.38	2.38

Tablo 4.5. TF-IDF hesabı

Cümle	Ayşe	Eve	Geldi	Koşarak
Ayşe eve geldi	0.693	0.693	0.693	0
Ayşe eve koşarak geldi	0.595	0.595	0.595	0.595

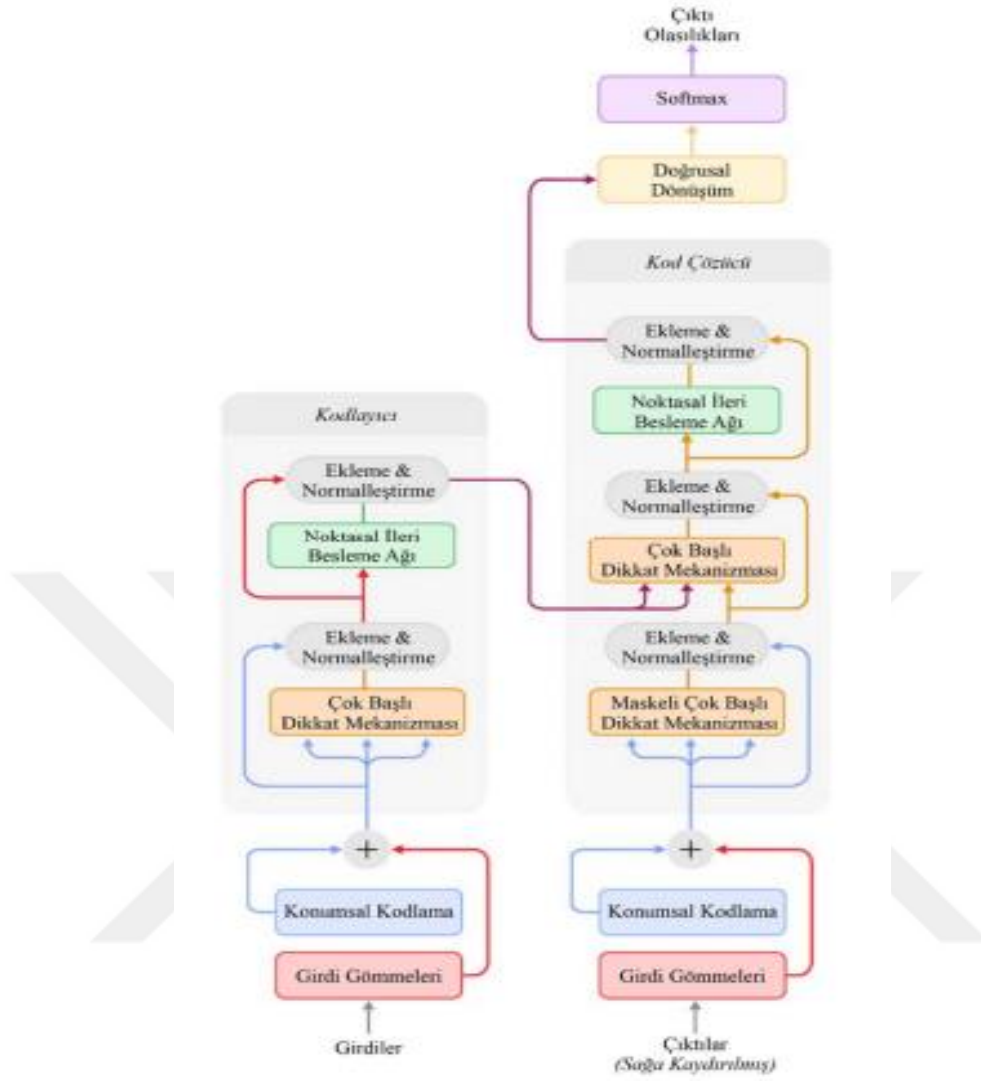
Count vectorizer, kelimelerin ilgili metindeki geçme frekansını ölçer. Bütün verideki eşsiz kelimeler çıkarılır. Daha sonrasında tek tek satırlara bu eşsiz kelimelerin her bir metinde geçip geçmediğine bakılır. Geçerse 1 geçmezse 0 verilir. Count vectorizer ile her bir kelime özellik niteliğinde kullanılıp sayısal ifadelere dönüşmüş olur. Böylelikle algoritmalarda bu kelime vektörleri girdi olarak kullanılır. Tablo 4.6’da count vectorizer hesaplama örneği gösterilmiştir. Metinde geçen her bir kelime çıkarılmıştır. Bu işleme tokenizasyon işlemi denilmektedir. Bir cümleyi kelimelerine ayırma işlemi yapılmıştır. “Elif ders çalışıyor” cümlesinde Elif kelimesinin geçtiği yere 1, geçmediği yere 0 aynı şekilde ders kelimesinin cümlede geçtiği yere 1 geçmediği yere 0 verilmiştir. Diğer kelimeler içinde aynı işlem uygulanarak sözel ifadeler matematiksel olarak sayısal ifadelere dönüştürülmüş olur. (Kaynar ve diğ., 2016).

Tablo 4.6. Count vectorizer hesaplama örneği

Cümleler	Aldı	Başarı	Belgesi	Çalışmak	Çalışmayı	Çalışıyor	Ders	Elif	Getirir	Seviyor
Elif ders çalışıyor	0	0	0	0	0	1	1	1	0	0
Elif ders çalışmayı seviyor	0	0	0	0	1	0	1	1	0	1
Ders çalışmak başarı getirir	0	1	0	1	0	0	1	0	1	0
Elif başarı belgesi aldı	1	1	1	0	0	0	0	1	0	0

4.2.4. Dönüştürücüler (Transformers)

Dönüştürücüler, Google tarafından 2017 yılında “Attention Is All You Need” makalesinde tanıtılmıştır. Dönüştürücülerde kodlayıcı (encoder) ve kod çözücü (decoder) kısımları bulunur. Kodlayıcı kısım bir metindeki her bir kelimenin diğer kelimelerle ilişkisine bakarak bir vektör oluşturur. Kod çözücü kısım ise kodlayıcı kısımdan aldığı vektörlerle hedef olan çıktıyı üretir. Dönüştürücü mimarisine ait akış Şekil 4.6’da gösterilmiştir (Karaca ve Aydın , 2024). Bu şekildeki kodlayıcı kısmına baktığımızda metindeki kelimeler vektörlerle ifade edilir. Daha sonra her bir kelimenin konumu bu vektörlere eklenir. Çok başlı dikkat mekanizması kısmında öz dikkat mekanizması (self attention) uygulanır. Dikkat mekanizması ile her kelimenin diğer kelimelerle olan ilişkisi hesaplanır. Daha sonra hesaplanan vektörlere normalleştirme işlemi uygulanır. Noktasal ileri besleme ağına gönderilir. Bu ağ katmanlardan çıkan sinyali sonraki kaymana iletir. Veriler ileri yönlü bir akışla işlem görür. Geri yönlü bir besleme görülmez. Sonrasında çıkan sonuç kod çözücü katmanına gönderilir. Kodlayıcı kısmı girdi olan metni işler anlamlı bir hale dönüştürür. Çözücü kısım ise kodlayıcı kısmını kullanarak çıktı üretir.



Şekil 4.6. Dönüştürücü mimarisi (Karaca ve Aydın , 2024)

4.2.5. BERT

Bidirectional encoder representations from transformers kelimelerinin ilk harflerinden oluşmuş olan BERT modeli Google tarafından 2018 yılında duyurulmuştur (Ozan ve diğ., 2021). Doğal dil işlemede her kelime anlamı tek yönlü yani soldan sağa olarak bakılır ve öğrenilir. BERT modelinde ise çift yönlü yani soldan sağa ve sağdan sola şeklinde bir öğrenme metodu mevcuttur. Bu nedenle her cümlelerin veya metnin tamamının doğru bir şekilde öğrenilmesini ve anlamlandırılmasını sağlar.

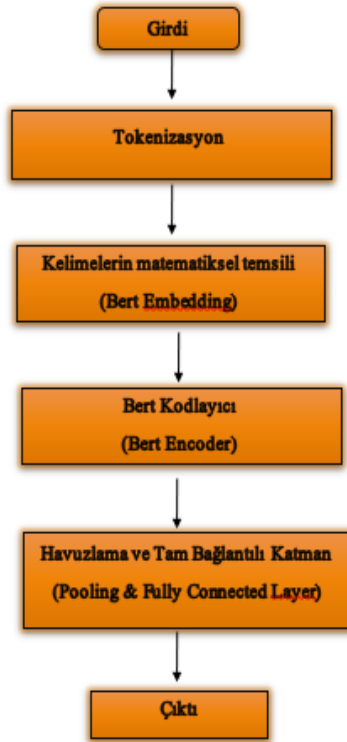
BERT, dönüştürücüler (transformers) mimarisine dayanan bir modeldir. BERT, bu dönüştürücünün kodlayıcı kısmını kullanır. Kodlayıcı kısımda her bir kelimenin diğer

kelimelerle arasındaki ilişkiye dikkat mekanizması (self attention) adı verilen bir modelleme kullanılır.

Şekil 4.7’de BERT modelinin akış şeması verilmiştir. Metin girdi şeklinde alınır. Tokenizasyon işlemi uygulanarak metin kelimelere ayrılır. Sonrasında her bir kelime vektörlerle ifade edilir. BERT kodlayıcı adımıyla ise öz dikkat mekanizması (self attention) devreye girer. Bu mekanizma ile her kelime arasındaki ilişki hesaplanır. Havuzlama aşamasında metindeki önemli verileri seçer ve verinin boyutunu düşürmüştür. Tam bağlantılı katmanda her vektör ağırlıklarla çarpılır. Relu veya softmax gibi aktivasyon fonksiyonları uygulanabilir. Çıkan sonuç sınıflandırmayı verir. Sonuç olarak örneğin olumlu veya olumsuz şeklinde bir çıktı elde edilebilir.

BERT önceden eğitim modeli olarak maskeli dil modelleme (masked language modeling) kullanarak eğitilmektedir. Bu modelde girdideki belirli kelimeler maskelenir. Model ise bu maskelenmiş olan kelimeyi bulmaya çalışır. Örneğin “Ali elma [MASK]” şeklinde maskelendiğinde “topladı” veya “yedi” kelimesini bulmaya çalışır. Maskelenmiş kelimeler cümledeki tüm kelimelerle ilişkili olduğu için çok yönlü dil anlayışına sahip olur. Eğitim aşamasında maskeli dil modelinden farklı olarak sonraki cümle tahmini (next sentence prediction) ile ön eğitim modeli yapılmaktadır. Bu model ard arda gelen iki cümlelerin anlam bakımından ard arda gelmesinin doğru olup olmadığını belirler. “Akşam işten çıkarım” ilk cümle ve ikinci cümle olarak “otobüsle eve giderim” gelmesi durumu anlam bakımından uyumlu ise pozitif uyumlu değil ise negatif olarak etiketlenir. Bu şekilde metindeki kelimeleri tahmin ederek kelimelerin bağlamını anlamaya çalışır.

BERT modeli çoklu dil desteği sağlamaktadır. Yalnızca İngilizce değil birçok dilde model geliştirilebilir. Duygu analizinde pozitif, negatif ve nötr şeklinde kategorine ayırma işlemlerinde, Türkçe’den İngilizce’ye gibi çevirilerde, bir kullanıcının sorduğu soruya en uygun ve doğru cevabı bulma gibi işlemlerde, metinlerin özetlerini çıkarma gibi işlemlerde, yarım kalmış bir metnin tamamlanması gibi işlemlerde kullanılabilir.



Şekil 4.7. BERT akış şeması

4.2.6. Öz Dikkat Mekanizması (Self Attention)

Kelimelerin her birinin birbirlerine olan uzaklıklarını ve ilişkisini öğrenmeyi sağlayan mekanizmadır. Metindeki her bir kelime eşit oranda öneme sahip değildir. Cümlelerin anlamına en çok etki eden kelime duygu durumuna daha çok etki edeceği için daha fazla ön plana çıkarılır (Onan, 2022). Girdi olarak her bir kelime vektörlerle ifade edilir.

Bu adımda query, key ve value vektörleri kullanılır. Sorgu (query) bir kelimenin diğer kelimelerle ne kadar ilgili olduğunu sorgulayan vektördür. Anahtar (key) vektörü ise kelimenin diğer kelimelerle eşleşmesi için kullanılır. Değer (value) vektörü ise anahtar ve sorgu vektörü arasındaki ilişkinin sonucunda hangi değeri verdiğini belirlemek için kullanılır. Örneğin Ali elma topladı cümlesinde Ali kelimesinin diğer kelimelerle ne kadar bağlantılı olduğuna bakılırsa sorgu vektörü Ali kelimesinin diğer kelimelerle ilişkisini sorgulamak için kullanılır. Ali kelimesinin elma ve topladı kelimeleriyle arasında nasıl bir ilişkisi olduğuna bakılır. Diğer anahtar kelimeler elma ve topladı. Ali kelimesi ile elma kelimesinin ve topladı kelimelerinin arasındaki nasıl ilişki olduğunu temsil eden anahtar vektördür. Ali kelimesinin bir özne, elma kelimesinin bir nesne ve

topladı kelimesinin fiil olması da değer vektörüdür. Her kelimenin içerik bilgisini temsil eder.

1. Adım: Kelimelerin vektörle ifade edilmesi işlemi yapılır. Bu vektörlerin sayısal değerleri rastgeledir. One-hot encoding, TF-IDF gibi değerlerle elde edilebilir.

Ali için [1,0,1]

Elma için [0,1,0]

Toplandı için [1,1,0]

2. Adım: Her kelime için sorgu, anahtar ve değer matrislerinin oluşturulması yapılır. Bu matris değerleri rastgeledir. Tablo 4.7 'de öz dikkat mekanizması sorgu, anahtar ve değer matrisleri tablosu verilmiştir. Denklem (4.11)'de sorgu matrisi, Denklem (4.12)'de anahtar matrisi ve Denklem (4.13)'te değer matrisi gösterim şekli yer almaktadır.

Tablo 4.7. Öz dikkat mekanizması sorgu,anahtar, değer matrisi

Kelime	Sorgu (Q)	Anahtar (K)	Değer (V)
Ali	[1,0,1]	[1,0,1]	[1,0,1]
Elma	[0,1,0]	[0,1,0]	[0,1,0]
Toplandı	[1,1,0]	[1,1,0]	[1,1,0]

$$\text{Sorgu matrisi}(Q) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.11)$$

$$\text{Anahtar Matrisi (K)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.12)$$

$$\text{Değer Matrisi (V)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.13)$$

3. Adım: Her kelime diğer kelimelerle olan ilişkisinin hesaplanması için sorgu ve anahtar vektör çarpımı yapılır. Her kelime için hesaplama yapıp skor matrisi oluşturulur.

Skor hesaplaması Denklem (4.14)'te verilmiştir. Denklem (4.15)'te K'nın transpozu, Denklem (4.16)'da Q değeri ve Denklem (4.17)'de skor hesabı sonucu verilmiştir.

$$Skor = Q * K^T \quad (4.14)$$

$$K^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (4.15)$$

$$Q = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad (4.16)$$

$$Q * K^T = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (4.17)$$

4. Adım: Softmax formülü ile çıkan sonuçlara normalizasyon işlemi uygulanır. Her sonucu 0 ile 1 arasında bir değere dönüştürür. Denklem (4.18)'de bir kelime için softmax formülü verilmiştir.

$$Softmax(x_i) = \frac{e^{(x_i)}}{\sum_{j=1}^n e^{x_j}} \quad (4.18)$$

Denklem (4.19)'da Ali kelimesi için dikkat vektörü hesaplaması verilmiştir.

$$Softmax(2,0,1) = \frac{e^2}{e^2+e^0+e^1}, \frac{e^0}{e^2+e^0+e^1}, \frac{e^1}{e^2+e^0+e^1} \quad (4.19)$$

Denklem (4.20)'de yaklaşık değerler hesaplanır;

$$Softmax(2,0,1) = [0.66, 0.009, 0.24] \quad (4.20)$$

5. Adım: Bulunan bu ağırlıklar değer vektörü ile çarpılır ve toplanır. Böylelikle Ali kelimesi için yeni vektör oluşturulmuş olur. Bu hesaplama ile her kelimenin birbiri ile bağlantılı vektörel temsilini vermektedir. Denklem (4.21)'de öz dikkat mekanizmasının

formülü verilmiştir. Denklem (4.22)'de Ali kelimesi için öz dikkat mekanizma hesaplaması verilmiştir.

$$Self - attention = Softmax \left(\frac{(Q * K^T)}{\sqrt{dk}} \right) * V \quad (4.21)$$

$$Softmax(2,0,1) = 0.66 * [1,0,1] + 0.09 * [0,1,0] + 0.24 * [1,1,0] \quad (4.22)$$

4.3. Performans Ölçüm Metrikleri

Makine öğreniminde model eğitilip tahminleme yapıldıktan sonra modelin performansını değerlendirmek için ölçüm metrikleri kullanılır. Bu metrikler bize modelin başarısını vermektedir. Modelin görevini ne kadar iyi gerçekleştirdiğinin görünmesini sağlar.

Doğruluk puanı (Accuracy Score): Acc (Doğruluk puanı), sınıflandırma yapıldıktan sonra doğru tahmin edilen sayının veri setindeki verilerin sayısına oranıdır. Örneğin bir veri setinde 100 adet veri varsa ve 90 tanesi doğru tahmin edildiyse bu modelin doğruluk oranı %90'dır. Denklem (4.23)'te Doğruluk hesaplama formülü verilmiştir. Formül hesaplanması, doğru pozitif sayısı ile doğru negatif sayısının toplamının doğru pozitif, doğru negatif, yanlış pozitif, yanlış negatif toplamına oranıdır.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.23)$$

TP (True Positive): Doğru pozitif sayısını temsil eder. Örneğin iyi olan veriye iyi tahmininde bulunulmuştur.

TN (True Negative): Doğru negatif sayısını temsil eder. Örneğin kötü olan bir değer kötü olarak tahminde bulunulmuştur.

FP (False Positive): Yanlış pozitif sayısını temsil eder. Örneğin kötü olan bir veri iyi olarak tahminlenmiştir.

FN (False Negative): Yanlış negatif sayısını temsil eder. Örneğin iyi olan bir veri kötü olarak tahminde bulunulmuştur.

Kesinlik puanı (Precision score): Pre (Kesinlik), sınıflandırma yapıldıktan sonra pozitif olarak tahmin edilenlerin sayısının kaç tanesinin gerçekte pozitif olduğunu gösteren

puandır. Denklem (4.24)'te kesinlik puanının hesaplama formülü verilmiştir. Formül hesaplanması doğru pozitif sayısının doğru pozitif ve yanlış pozitifin toplamına oranıdır.

$$Pre = \frac{TP}{TP+FP} \quad (4.24)$$

TP (True Positive): Doğru pozitif sayısını temsil eder.

FP (False Positive): Yanlış pozitif sayısını temsil eder.

Duyarlılık puanı (Recall score): Rec (Duyarlılık puanı), tahmin edilen pozitiflerin sayısının ne kadarının doğru tahmin edildiğini gösteren puandır. Denklem (4.25)'te duyarlılık hesaplama formülü verilmiştir. Toplam pozitif sayısının toplam pozitif ve yanlış negatif toplamına bölümü ile hesaplanır (Özluer ve diğ., 2021).

$$Rec = \frac{TP}{TP+FN} \quad (4.25)$$

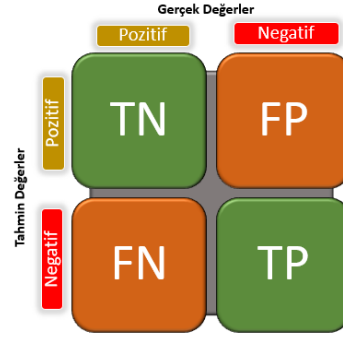
TP (True Positive): Doğru pozitif sayısını temsil eder.

FN (False Negative): Yanlış negatif sayısını temsil eder.

F1 puanı (F1 score): Kesinlik ve duyarlılık puanları arasında denge bulunmaktadır. F1 puanı duyarlılık ve kesinlik puanlarının harmonik ortalamasıyla hesaplanır. Denklem (4.26)'da F1 puan hesaplama formülü verilmiştir. F1 puanı kesinlik ve duyarlılık yüksek olduğunda yüksek bir değere sahip olur (Küsmüş, 2024).

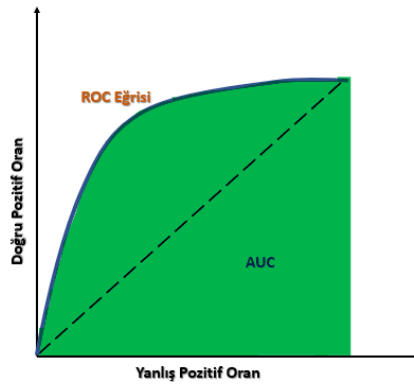
$$F1 \text{ puanı} = 2 * \frac{\text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (4.26)$$

Karışıklık matrisi: Kullanılan modelin bulduğu verilerle gerçek verilerin performansını değerlendiren ve onları tablo formatında sunan bir matristir. Algoritmadan çıkan sonuca göre tahminleme dağılımını özet şeklinde bir tabloda sunar. Makine öğrenmesi ve istatistik alanında sıklıkla kullanılan hata analiz aracıdır. Algoritmanın hangi alanda hata yaptığını gösterir. İki boyutlu olan bu matrisin satırında gerçek değerler sütununda ise tahmin edilen değerler bulunmaktadır. Doğru ve yanlış tahmin edilmiş değerlerin yüzdesini bulmak için kullanışlıdır (Heydarian ve diğ., 2022). Şekil 4.8 'de ikili bir sınıflandırma için karışıklık matrisi gösterilmektedir.



Şekil 4.8. İkili sınıflandırma için karışıklık matrisi yapısı

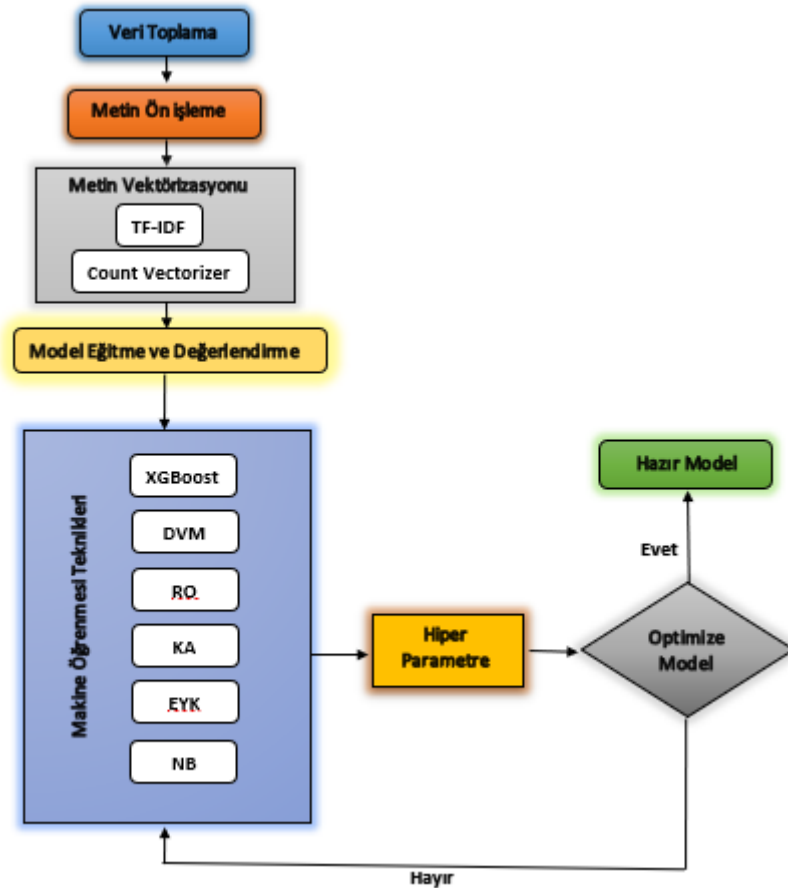
ROC analizi: ROC eğrisi y ekseninde doğru pozitif, x ekseninde ise yanlış pozitif oranlarını veren eğridir. (Tomak ve Bek, 2010) Doğru pozitiflik oranı ile yanlış pozitif oranı orantılıdır. Doğru pozitif oranı düşerse yanlış pozitif oranı da düşer. En iyi ROC eğrisini bulmak için doğru eşik değerini bulmak gerekir. Örneğin pozitif ve negatif olarak sınıflandırılacak olan veri setinde yapılan tahminler eşik değer ile karşılaştırılır. Eşik değerden büyükse pozitif küçükse negatif olarak sınıflandırılır. Yapılan tahminler sonucunda F1 puanlar hesaplanmaktadır. Bu puanlardan en iyisi bulunduğu en iyi ROC eğrisi oluşmuş demektir. ROC eğrisinin altında kalan alan ise AUC değerini temsil eder. AUC değeri 0 ile 1 arasında değişir. Düşük AUC değeri modelin performansının düşük olduğu anlamına gelmektedir. AUC değeri 1 ise tüm sınıflandırma modelinde hiç yanlış sınıflandırma yapmadığı anlamına gelmektedir. AUC değerinin 0 olması ise tüm sınıflandırma modelinde hiç doğru bir sınıflandırma yapmamış demektir. Şekil 4.9’da ROC analizi örnek şekli yer almaktadır.



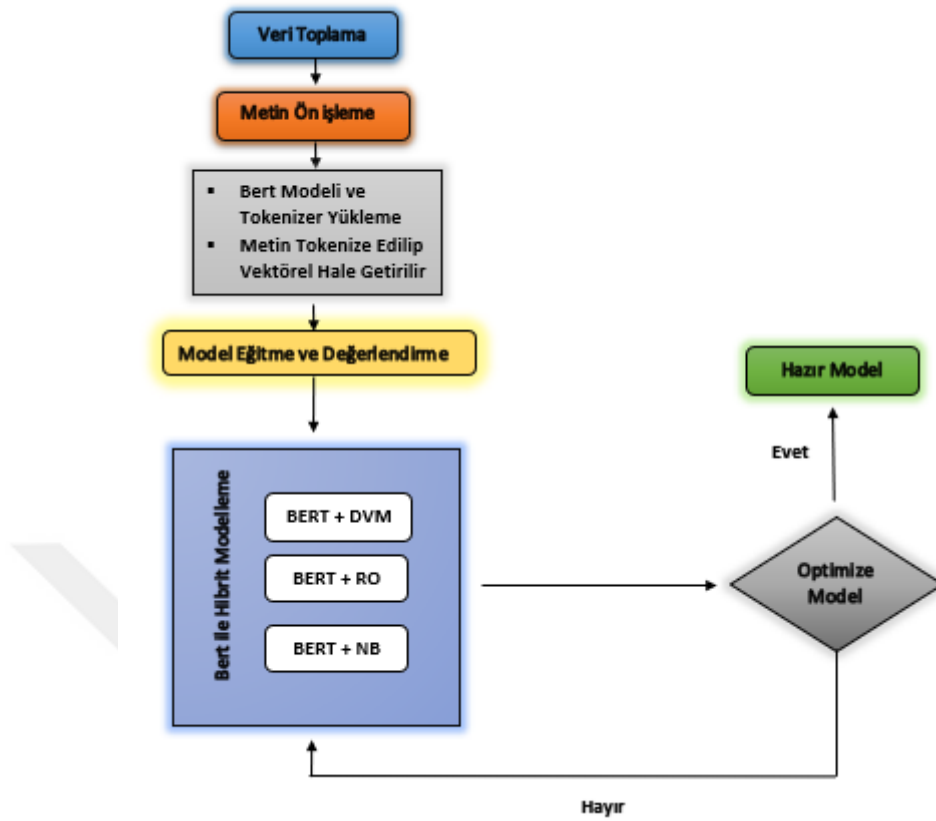
Şekil 4.9. ROC analizi örneği

5. DENEYSEL ÇALIŞMALAR

Bu araştırmada makine öğrenmesi algoritmaları kullanılarak yapılan çalışma Şekil 5.1’de gösterildiği gibi özetlenmiştir. Şekil 5.2’de ise BERT ve BERT ile hibrit çalışan modellemenin akış şeması gösterilmiştir. Veri toplama aşamasında herkese açık olan veriler Kaggle platformundan elde edilmiştir. Metin ön işleme aşamasında metnin kalitesini artırmak ve modellemek için uygun hale getirilmiştir. Metin vektörizasyonu aşamasında TF-IDF ve count vectorizer kullanılarak veri setindeki makine öğrenmesi modellerinin anlayabileceği sayısal ifadelere dönüştürülmüştür. Daha sonrasında eğitim ve test olarak veri seti ayrılmıştır. DVM, RO, KA, XGBoost, EYK, GNB, MNB, BERT, BERT+DVM, BERT+GNB, BERT+RO algoritmaları uygulanmıştır. EYK, XGBoost, RO ve DVM algoritmalarına hiper parametre optimizasyonu uygulanır. Bütün modellerden çıkan sonuçlara göre en iyi olan yöntem belirlenir.



Şekil 5.1. Önerilen makine öğrenmesi temelli yöntemin akış şeması



Şekil 5.2. Önerilen BERT ile hibrit modelleme temelli yöntemin akış şeması

5.1. Veri seti

Luca Mongelli tarafından hazırlanan veri seti Kaggle 'dan elde edilmiştir (URL-4). 5868 satır bulunan bu veri setinde Id, polarity ve text alanları mevcuttur. Polarity alanında 'positive', 'negative' ve 'neutral' özellikleri bulunmaktadır. Bu verilerin 1128 tanesi pozitif ile, 786 tanesi negatif ile, 3954 tanesi ise nötr ile etiketlidir. Id alanı metinlerin Id'sidir ve metin sütunu ise projede asıl uygulamanın yapılacağı kısımdır. Tablo 5.1' de pozitif sınıfa etiketlenmiş olan veri setindeki örnek cümleler yer almaktadır. Tablo 5.2'de negatif sınıfa etiketlenmiş olan veri setindeki örnek cümleler yer almaktadır. Tablo 5.3'te tarafsız sınıfa etiketlenmiş olan veri setindeki örnek cümleler yer almaktadır. Bu veri seti herkesin kullanımına açık bir veri setidir. Thanks gibi kelimelerin cümlede pozitif olduğu belirgin olduğundan genellikle pozitif kategorisinde yer almaktadır. Sorry gibi cümlenin anlamını olumsuz etkileyen kelimeler ise cümleye olumsuzluk anlamı verdiğinden genellikle olumsuz kategorisinde yer almaktadır.

Tablo 5.1. Veri setinde bulunan pozitif sınıfa ait öznitelik örneği

Yorum	Sınıf
Thanks Alexey	Pozitif
Looks good to me.	Pozitif
John it would be great!	Pozitif
Fix applied thanks.	Pozitif
Excellent thanks a lot!	Pozitif
great idea those are lost to the ages but I'll try to reproduce.	Pozitif
Automatic location selection is very cool.	Pozitif

Tablo 5.2. Veri setinde bulunan negatif sınıfa ait öznitelik örneği

Yorum	Sınıf
Sorry about that.	Negatif
I didn't have time to work on this yet ...	Negatif
If I see it correctly	Negatif
Oops sorry folks. Wrong project.	Negatif
Who made this stupid rule?	Negatif
I think it's time to just close this issue.	Negatif
What the hell are you waiting for?	Negatif

Tablo 5.3. Veri setinde bulunan nötr sınıfa ait öznitelik örneği

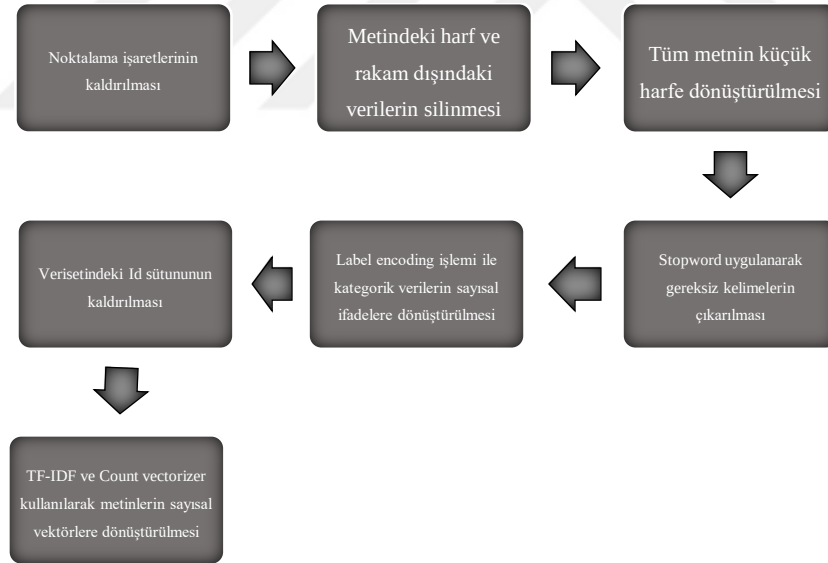
Yorum	Sınıf
See comments above	Nötr
Committed in rev 983396	Nötr
Closing resolved issues	Nötr
Test log	Nötr
Duplicate of XERCESS-1810	Nötr
Fixed typo in title	Nötr
Changes look reasonable to me.	Nötr

5.2. Veri Ön İşleme

Veri setinde çeşitli ön işleme çalışmaları yapılmıştır. Id sütunu projenin geliştirilmesinde etkisiz olacağı için silinmiştir. Polarity alanı metin olduğu için label encoding işlemi uygulanarak sayısal ifadeye çevrilmiştir. Negatif olan sınıflandırma kategorisi 0 olarak,

nötr 1 olarak ve 2 pozitif olarak sayısal verilere dönüştürülmüştür. Harflerden ve rakamlardan (A-Z/a-z/0-9) farklı olan karakterlerin hepsi silinmiştir. Harflerin tamamı küçük harfe çevrilmiştir. Noktalama işaretleri kaldırılmıştır. Metin alanında geçen bazı kelimelerin cümlelerin olumlu veya olumsuz olmasına bir etkisi yoktur. Örnek olarak “this” kelimesi verilebilir. Bunun gibi etkisi olmayacak kelimelerin veri setindeki metin alanından natural language toolkit (nltk) doğal dil işleme kütüphanesi indirilerek ve stopwords kullanarak çıkarılmıştır. I, we, you, and, the, an, some gibi kelimeler stopwords kelimelere örnektir. Stopwords kelimelerinin içinde İngilizce olumsuzluk ifade eden “not” kelimesi vardır. Bu kelime çıkarıldığında cümle olumlu bir cümleye döndüğü için “not” kelimesi stopwords içinden çıkarılmıştır.

Bu çalışmada 2 çeşit sayısal ifadeye dönüştürme yöntemi kullanılmıştır. Bunlar count vectorizer ve TF-IDF (Term Frequency-Inverse Document Frequency) yöntemleridir. Şekil 5.3’ te yapılan ön işlemenin akış şeması gösterilmektedir. Şekil 5.4’de TF-IDF uygulamasından sonra veri setinin nasıl görüldüğü gösterilmektedir.



Şekil 5.3. Veri ön işleme akış şeması

	103	104	105	106	107	108	109	110	111	112	113	114	115
1244	0	0	0	0	0	0	0	0	0	0	0	0	0
1245	0	0	0	0	0	0	0	0	0	0	0	0	0
1246	0	0	0	0	0	0	0	0	0	0	0	0	0
1247	0	0	0	0	0	0	0	0	0	0	0	0	0
1248	0	0	0	0	0	0	0	0	0	0	0	0	0
1249	0	0	0	0	0	0	0	0	0	0	0	0	0
1250	0	0	0	0	0	0	0	0	0	0	0	0	0
1251	0	0	0	0	0	0	0	0	0	0	0	0	0
1252	0	0	0	0	0	0	0	0	0	0	0	0	0
1253	0	0	0	0	0	0	0	0	0	0	0	0	0
1254	0	0	0	0	0	0	0	0	0	0	0	0	0
1255	0	0.219173	0	0	0	0	0	0	0	0	0	0	0
1256	0	0	0	0	0	0	0	0	0	0	0	0.691695	0
1257	0	0	0	0	0	0	0	0	0	0	0	0	0
1258	0	0	0	0	0	0	0	0	0	0	0	0	0
1259	0	0	0	0	0	0	0	0	0	0	0	0	0
1260	0	0	0	0	0	0	0	0	0	0	0	0	0
1261	0	0	0	0	0	0	0	0	0	0	0	0	0
1262	0	0	0	0	0	0	0	0	0	0	0	0	0

Şekil 5.4. TF-IDF uygulanması sonucu veri setinin görünümü

5.3. Hiperparametre Optimizasyonu

Bu çalışmada 5868 satır Jira yorumu içeren veri seti kullanılmıştır. Jira'daki yorumların pozitif negatif ve nötr olarak sınıflandırılması yapılmıştır Programlama dili olarak Python kullanılmıştır ve kodlar Spyder geliştirme ortamında yazılmıştır. Veri seti Kaggle ortamından elde edilmiştir. Veri setinin %80'i eğitim %20'si test seti olarak ayrılmıştır. Veri setinde bulunan metin alanlarının sayısallaştırılması için TF-IDF ve count vectorizer yöntemleri kullanılmıştır. XGBoost, EYK, DVM, RO, KA, GNB, MNB, BERT, BERT+DVM, BERT+GNB, BERT+RO algoritmaları kullanılarak sınıflandırma işlemi yapılmıştır.

BERT modelinin hibrit çalışması veri ön işlemeden sonra her bir kelime sayısal veriye dönüştürülür. Bu işleme tokenizer denir. Tokenize edilmiş kelimeler BERT modeline gönderilir ve her kelime için matematiksel vektör olan embedding vektörüne dönüştürülür. Her kelimenin ortalaması alınır. Bu ortalamalar ile metni temsil eden vektör elde edilir. Sayısal hale çevriler metinler DVM, GNB ve RO modelleriyle tahminleme işlemlerini gerçekleştirir.

XGBoost, DVM, RO, EYK algoritmalarına hiper parametre optimizasyonu uygulanmıştır. Tablo 5.4'te TF-IDF yöntemine göre DVM, EYK, XGBoost ve RO algoritmalarına yapılan hiper parametreler yazılmıştır. En başarılı parametreler tabloda koyu renk ile yer almaktadır. Hiper parametre optimizasyonu yapmak için sklearn

kütüphanesinde bulunan GridSearchCV yöntemi kullanılmıştır. 3 katmanlı çapraz doğrulama işlemi yapılmıştır. Çapraz doğrulamada veri setini 3 parçaya böler ve sırasıyla her bir parçayı test seti olarak diğer kalan iki parçayı ise eğitim seti olarak kullanır. Model eğitim setinde eğitilip test setinde performansı ölçülür. Her katman sonunda modelin performansı değerlendirilmiş olur. En yüksek performans hangi çapraz doğrulamada alındı ise bu işleme göre hiper parametreler en iyi parametre olarak seçilir.

Tablo 5.4. Hiper parameter optimizasyonu sonucu en iyi parametreler

Sınıflandırmalar	Hiperparametreler
DVM	kernel={' linear ', 'rbf'}, C={0.0001, 1 , 10}, gamma={ 1 , 10, 100}
EYK	n_neighbors={ 1 ,2,3,4,5,6,7,8,9}
XGBoost	max_depth={3,5, 7 }, learning_rate={ 0.1 , 0.01, 0.001}, subsample={ 0.5 , 0.7, 1}
RO	max_depth={ 80 , 90, 100, 110}, max_features={ 2 ,3}, min_samples_leaf={ 1 ,2,3,4,5,6,7,8,9,10}, min_samples_split={1, 6 ,11}, n_estimators={ 100 , 200, 300,500,1000}

6. DENEYSEL SONUÇLAR

Tablo 6.1’de count vectorizer kullanılarak alınan doğruluk, kesinlik, duyarlılık ve F1 puan performans sonuçları gösterilmiştir. Count vectorizer kullanılarak alınan sonuçlara bakıldığında en iyi sonucu veren algoritma doğruluk değeri (Acc) değeri %85,5 kesinlik (Pre) değeri %85,9 duyarlılık (Rec) değeri %85,5 ve F1 puanı değeri %85,2 ile XGBoost modelidir. Daha sonrasında sırasıyla DVM, RO, KA, EYK, MNB ve en kötü sonuç olan GNB algoritmasıyla devam etmektedir. En kötü sonucu veren GNB algoritmasının doğruluk değeri (Acc) değeri %50,3 kesinlik (Pre) değeri %62,7 duyarlılık (Rec) değeri %50,3 ve F1 puanı değeri %53,8 olarak tespit edilmiştir.

Tablo 6.1. Count vectorizer değerlendirme metrikleri

Sınıflandırma Modelleri	Performans Metrikleri (%)			
	Acc	Pre	Rec	F1-Puanı
GNB	50,3	62,7	50,3	53,8
DVM	82,5	83,1	82,5	81,5
KA	80,4	79,9	80,4	79,9
RO	80,8	80,5	80,8	80,0
EYK	77,7	77,3	77,7	76,3
XGBoost	85,5	85,9	85,5	85,2
MNB	73,7	74,4	73,7	73,6

Şekil 6.1’de test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi XGboost modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 95 tanesi doğru tahmin edilmiş, 60 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 709 tanesi doğru tahmin edilmiş, 11 tanesi negatif tahmin edilmiş, 65 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 200 tanesi doğru tahmin edilmiş, 1 tanesi negatif, 32 tanesi nötr tahmin edilmiştir.

Şekil 6.2’de test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi DVM modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 66 tanesi doğru tahmin edilmiş, 88 tanesi nötr tahmin edilmiş, 2 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 711 tanesi doğru tahmin edilmiş, 8

tanesi negatif tahmin edilmiş, 66 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 192 tanesi doğru tahmin edilmiş, 0 adet negatif, 41 tanesi nötr tahmin edilmiştir.

Şekil 6.3'te test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi RO modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 93 tanesi doğru tahmin edilmiş, 60 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 720 tanesi doğru tahmin edilmiş, 20 tanesi negatif tahmin edilmiş, 45 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 136 tanesi doğru tahmin edilmiş, 0 tanesi negatif, 97 tanesi nötr tahmin edilmiştir.

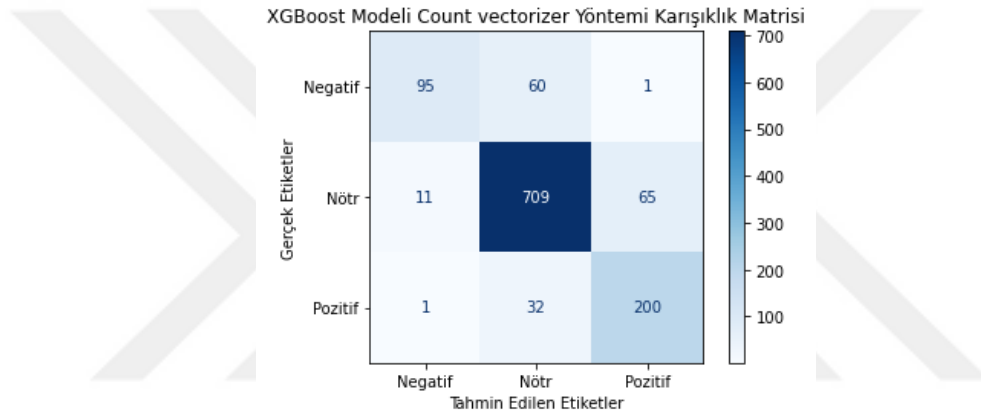
Şekil 6.4'te test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi KA modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 102 tanesi doğru tahmin edilmiş, 53 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 700 tanesi doğru tahmin edilmiş, 38 tanesi negatif tahmin edilmiş, 47 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 142 tanesi doğru tahmin edilmiş, 3 tanesi negatif, 88 tanesi nötr tahmin edilmiştir.

Şekil 6.5'te test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi EYK modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 71 tanesi doğru tahmin edilmiş, 84 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 720 tanesi doğru tahmin edilmiş, 20 tanesi negatif tahmin edilmiş, 45 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 122 tanesi doğru tahmin edilmiş, 0 tanesi negatif, 111 tanesi nötr tahmin edilmiştir.

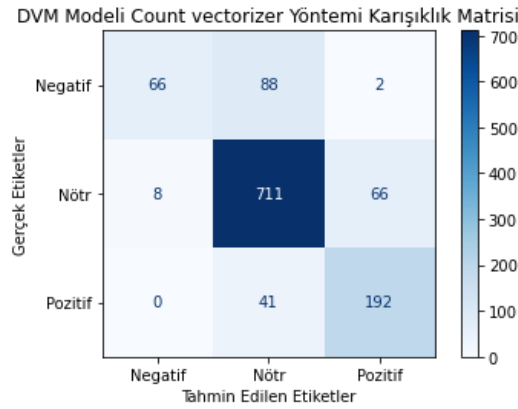
Şekil 6.6'da test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi GNB modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 89 tanesi doğru tahmin edilmiş, 56 tanesi nötr tahmin edilmiş, 11 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 416 tanesi doğru tahmin edilmiş, 238 tanesi negatif tahmin edilmiş, 131 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına

bakıldığında ise 86 tanesi doğru tahmin edilmiş, 88 tanesi negatif, 59 tanesi nötr tahmin edilmiştir.

Şekil 6.7’de test setine göre tahmin ve gerçek değerlerin count vectorizer yöntemi MNB modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 68 tanesi doğru tahmin edilmiş, 86 tanesi nötr tahmin edilmiş, 2 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 620 tanesi doğru tahmin edilmiş, 38 tanesi negatif tahmin edilmiş, 127 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 178 tanesi doğru tahmin edilmiş, 3 tanesi negatif, 52 tanesi nötr tahmin edilmiştir.

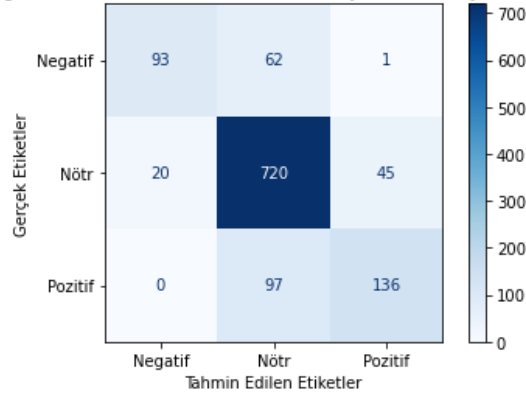


Şekil 6.1. Count vectorizer yöntemi XGBoost modeli karışıklık matrisi



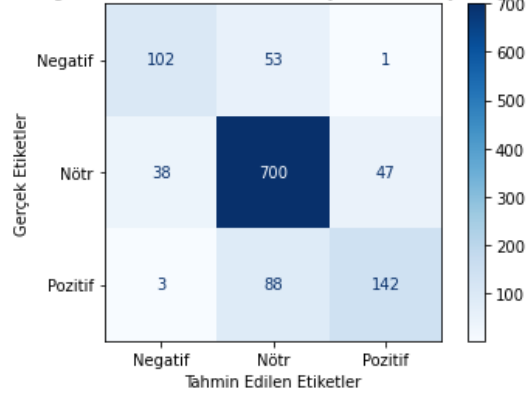
Şekil 6.2. Count vectorizer yöntemi DVM modeli karışıklık matrisi

Rastgele Orman modeli Count vectorizer yöntemi Karışıklık Matrisi



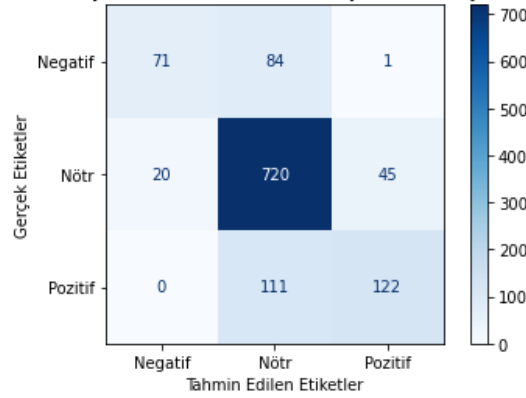
Şekil 6.3. Count vectorizer yöntemi RO modeli karışıklık matrisi

Karar Ağacı modeli Count vectorizer yöntemi Karışıklık Matrisi

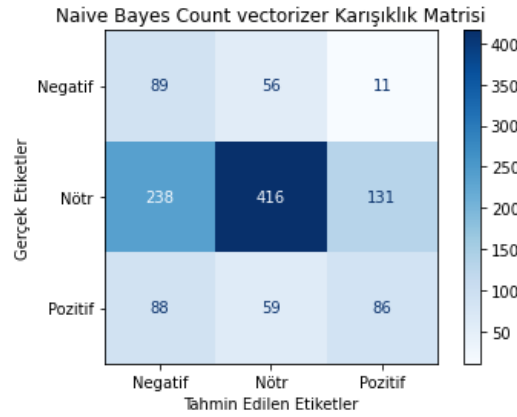


Şekil 6.4. Count vectorizer yöntemi KA modeli karışıklık matrisi

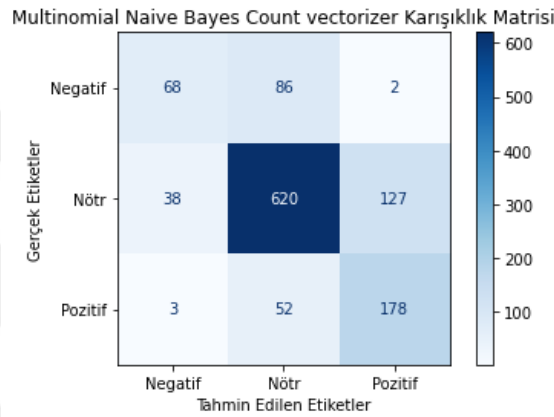
En yakın komşu modeli Count vectorizer yöntemi Karışıklık Matrisi



Şekil 6.5. Count vectorizer yöntemi EYK modeli karışıklık matrisi



Şekil 6.6. Count vectorizer yöntemi GNB modeli karışıklık matrisi



Şekil 6.7. Count vectorizer yöntemi MNB modeli karışıklık matrisi

Şekil 6.8’de count vectorizer yöntemi, XGboost modeli kullanılarak çizilmiş ROC analizi bulunmaktadır. Çizilen grafikte ROC eğrisi doğru pozitif oranı olan sol üst köşeye ne kadar yakın ise ve AUC değeri ne kadar yüksekse modelin performansı o oranda iyi olduğunu gösterir. Negatif sınıflandırma için AUC değeri %91, nötr değeri %90 ve pozitif değeri %95 olarak tespit edilmiştir.

Şekil 6.9’de count vectorizer yöntemi, DVM modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %91, nötr değeri %86 ve pozitif değeri %92 olarak tespit edilmiştir.

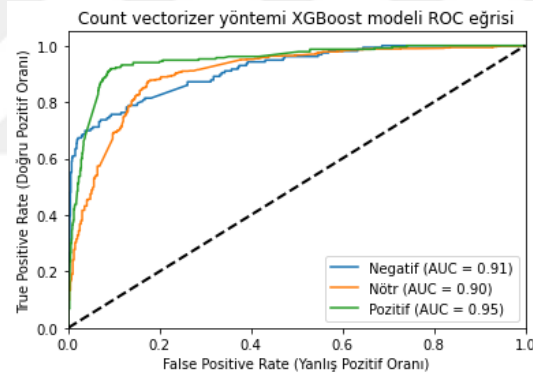
Şekil 6.10’da count vectorizer yöntemi, RO modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %87, nötr değeri %86 ve pozitif değeri %92 olarak tespit edilmiştir.

Şekil 6.11’de count vectorizer yöntemi, KA modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %81, nötr değeri %80 ve pozitif değeri %85 olarak tespit edilmiştir.

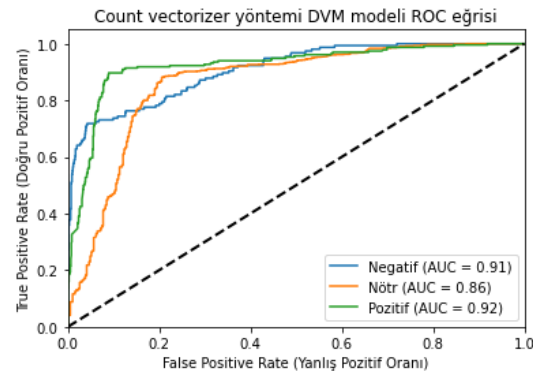
Şekil 6.12’de count vectorizer yöntemi, EYK modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %80, nötr değeri %80 ve pozitif değeri %87 olarak tespit edilmiştir.

Şekil 6.13’de count vectorizer yöntemi, GNB modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %63, nötr değeri %62 ve pozitif değeri %61 olarak tespit edilmiştir.

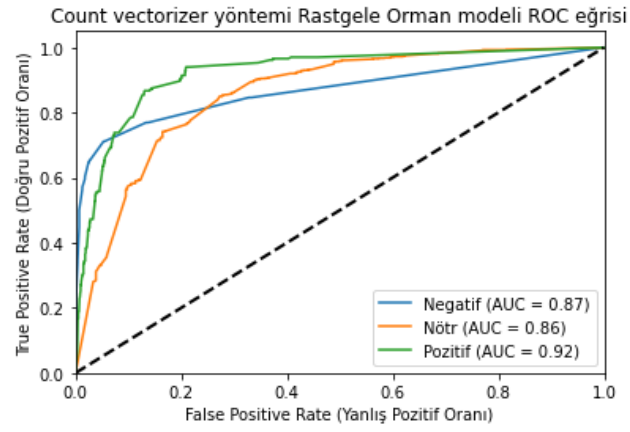
Şekil 6.14’te count vectorizer yöntemi, MNB modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %84, nötr değeri %77 ve pozitif değeri %87 olarak tespit edilmiştir.



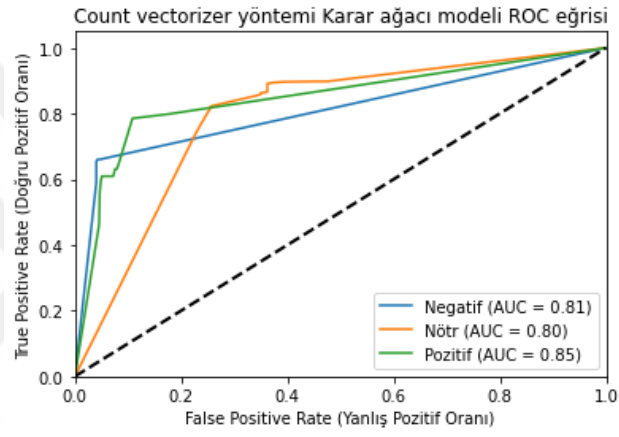
Şekil 6.8. Count vectorizer yöntemi XGBoost modeli ROC analizi



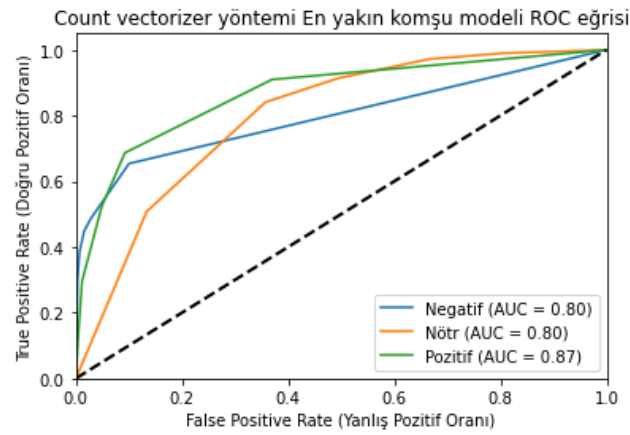
Şekil 6.9. Count vectorizer yöntemi DVM modeli ROC analizi



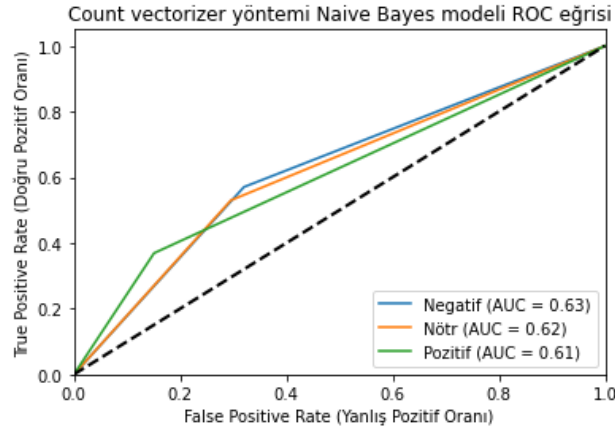
Şekil 6.10. Count vectorizer yöntemi RO modeli ROC analizi



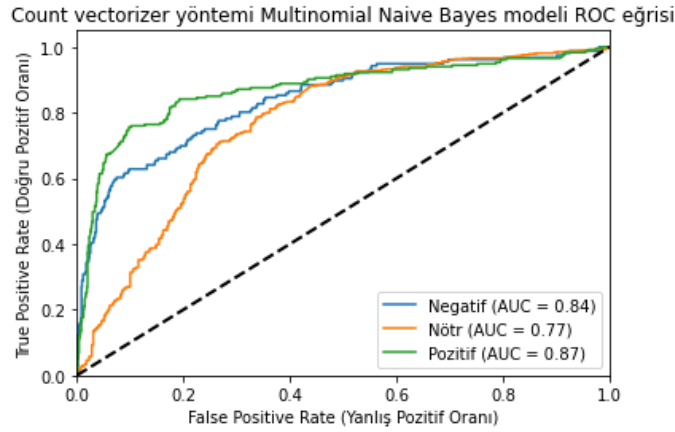
Şekil 6.11. Count vectorizer yöntemi KA modeli ROC analizi



Şekil 6.12. Count vectorizer yöntemi EYK modeli ROC analizi



Şekil 6.13. Count vectorizer yöntemi GNB modeli ROC analizi



Şekil 6.14. Count vectorizer yöntemi MNB modeli ROC analizi

Tablo 6.2’de TF-IDF kullanılarak alınan doğruluk, kesinlik, duyarlılık ve F1 puan performans sonuçları gösterilmiştir. TF-IDF kullanılarak alınan sonuçlara bakıldığında en iyi sonucu veren algoritma doğruluk değeri (Acc) değeri %86,4 kesinlik (Pre) değeri %86,7 duyarlılık (Rec) değeri %86,4 ve F1 puan değeri %86,0 ile XGBoost modelidir. Daha sonrasında sırasıyla DVM, KA, RO, EYK, MNB ve en kötü sonuç olan GNB algoritmasıyla devam etmektedir. En kötü sonucu veren GNB algortimasının doğruluk değeri (Acc) değeri %50,5 kesinlik (Pre) değeri %62,2 duyarlılık (Rec) değeri %50,5 ve F1 puan değeri %53,8 olarak tespit edilmiştir. Count vectorizer yöntemine göre değerler yakın olsa da TF-IDF yönteminde değerler biraz daha iyi sonuç vermiştir.

Tablo 6.2. TF-IDF değerlendirme metrikleri

Sınıflandırma Modelleri	Performans Metrikleri (%)			
	Acc	Pre	Rec	F1-Puanı
GNB	50,5	62,2	50,5	53,8
DVM	82,6	83,5	82,6	81,5
KA	84,5	84,5	84,5	84,5
RO	84,4	84,7	84,4	84,0
EYK	73,2	76,0	73,2	67,5
XGBoost	86,4	86,7	86,4	86,0
MNB	71,8	76,8	71,8	64,6

Şekil 6.15'te test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi XGboost modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 95 tanesi doğru tahmin edilmiş, 60 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 732 tanesi doğru tahmin edilmiş, 5 tanesi negatif tahmin edilmiş, 48 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 188 tanesi doğru tahmin edilmiş, 1 tanesi negatif, 44 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme aynı, nötr sınıfta TF-IDF yönteminde daha fazla, pozitif sınıfta ise daha az doğru tahmin yapıldığı tespit edilmiştir.

Şekil 6.16'da test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi DVM modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 74 tanesi doğru tahmin edilmiş, 82 tanesi nötr tahmin edilmiş, 0 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 742 tanesi doğru tahmin edilmiş, 2 tanesi negatif tahmin edilmiş, 41 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 154 tanesi doğru tahmin edilmiş, 0 adet negatif, 79 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme daha iyi, nötr sınıfta daha fazla, pozitif sınıfta ise daha fazla az tahmin yapıldığı tespit edilmiştir.

Şekil 6.17'de test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi RO modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 90 tanesi doğru tahmin edilmiş, 65 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 728 tanesi doğru tahmin edilmiş, 8 tanesi negatif

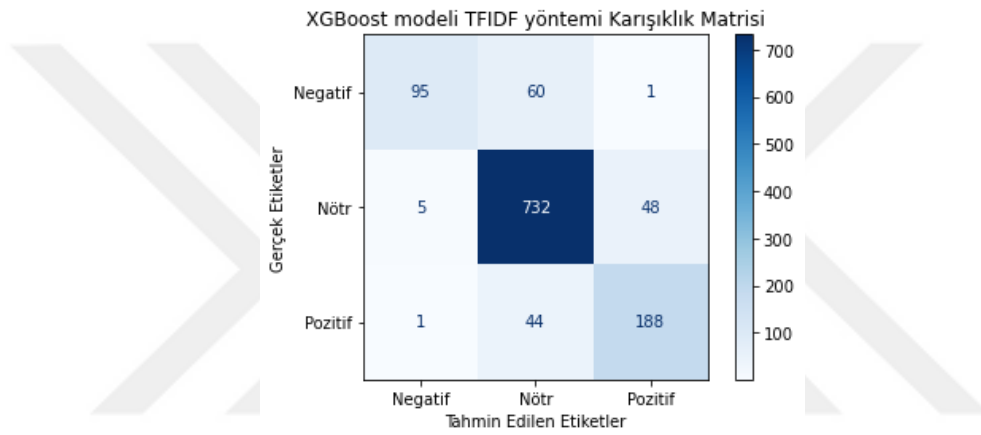
tahmin edilmiş, 49 tanesi pozitif tahmin edilmiştir. Pozitif sınıfa bakıldığında ise 174 tanesi doğru tahmin edilmiş, 0 adet negatif, 59 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme daha az, nötr sınıfta daha iyi, pozitif sınıfta ise daha fazla doğru tahmin yapıldığı tespit edilmiştir.

Şekil 6.18’de test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi KA modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 105 tanesi doğru tahmin edilmiş, 49 tanesi nötr tahmin edilmiş, 2 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfa bakıldığında 697 tanesi doğru tahmin edilmiş, 31 tanesi negatif tahmin edilmiş, 57 tanesi pozitif tahmin edilmiştir. Pozitif sınıfa bakıldığında ise 191 tanesi doğru tahmin edilmiş, 4 tanesi negatif, 38 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme daha iyi, nötr sınıfta daha az, pozitif sınıfta ise daha fazla doğru tahmin yapıldığı tespit edilmiştir.

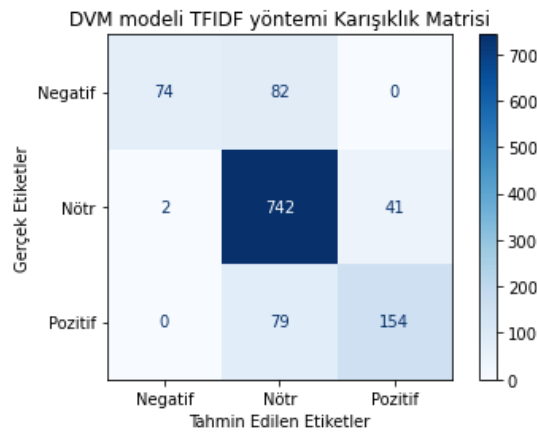
Şekil 6.19’da test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi EYK modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 41 tanesi doğru tahmin edilmiş, 114 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfa bakıldığında 771 tanesi doğru tahmin edilmiş, 1 tanesi negatif tahmin edilmiş, 13 tanesi pozitif tahmin edilmiştir. Pozitif sınıfa bakıldığında ise 48 tanesi doğru tahmin edilmiş, 2 tanesi negatif, 183 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme daha az, nötr sınıfta daha iyi, pozitif sınıfta ise daha az doğru tahmin yapıldığı tespit edilmiştir.

Şekil 6.20’de test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi GNB modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 88 tanesi doğru tahmin edilmiş, 57 tanesi nötr tahmin edilmiş, 11 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfa bakıldığında 418 tanesi doğru tahmin edilmiş, 220 tanesi negatif tahmin edilmiş, 147 tanesi pozitif tahmin edilmiştir. Pozitif sınıfa bakıldığında ise 88 tanesi doğru tahmin edilmiş, 84 tanesi negatif, 61 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre negatif sınıf için tahminleme daha az, nötr sınıfta daha fazla, pozitif sınıfta ise daha iyi doğru tahmin yapıldığı tespit edilmiştir.

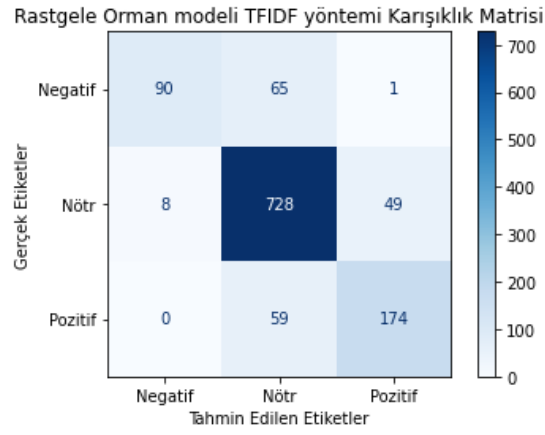
Şekil 6.21’de test setine göre tahmin ve gerçek değerlerin TF-IDF yöntemi MNB modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 26 tanesi doğru tahmin edilmiş, 130 tanesi nötr tahmin edilmiş, 0 adet ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 776 tanesi doğru tahmin edilmiş, 0 adet negatif tahmin edilmiş, 9 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 41 tanesi doğru tahmin edilmiş, 0 adet negatif, 192 tanesi nötr tahmin edilmiştir. Count vectorizer yöntemine göre TF-IDF yönteminde negatif sınıf için tahminleme daha fazla, nötr sınıfta daha fazla, pozitif sınıfta ise daha düşük doğru tahmin yapıldığı tespit edilmiştir.



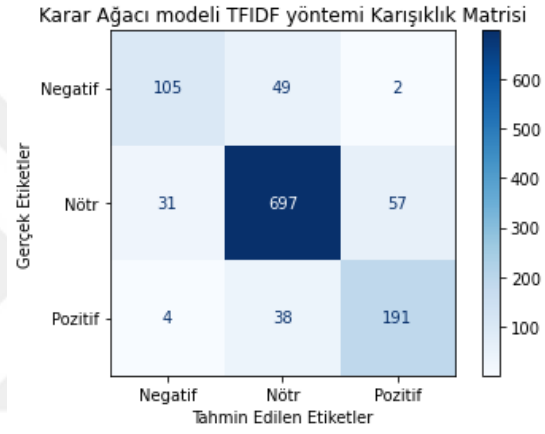
Şekil 6.15. TF-IDF yöntemi XGBoost modeli karışıklık matrisi



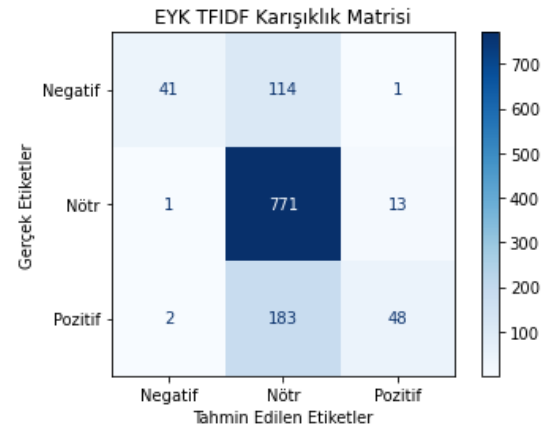
Şekil 6.16. TF-IDF yöntemi DVM modeli karışıklık matrisi



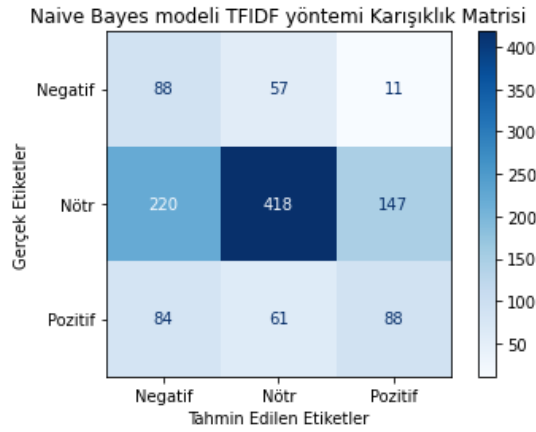
Şekil 6.17. TF-IDF yöntemi RO modeli karışıklık matrisi



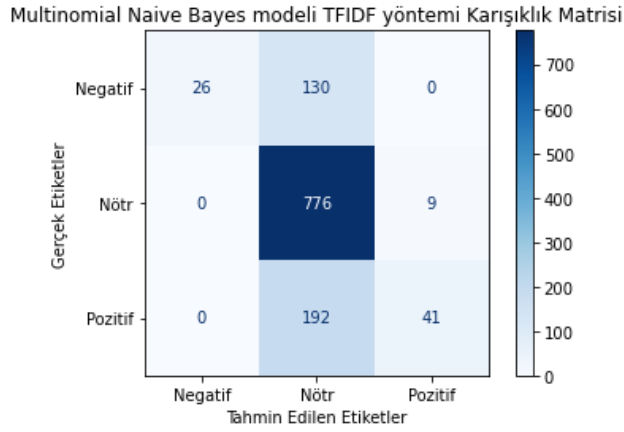
Şekil 6.18. TF-IDF yöntemi KA modeli karışıklık matrisi



Şekil 6.19. TF-IDF yöntemi EYK modeli karışıklık matrisi



Şekil 6.20. TF-IDF yöntemi GNB modeli karışıklık matrisi



Şekil 6.21. TF-IDF yöntemi MNB modeli karışıklık matrisi

Şekil 6.22’de TF-IDF yöntemi, XGboost modeli kullanılarak çizilmiş ROC analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %90, nötr değeri %90 ve pozitif değeri %95 olarak tespit edilmiştir.

Şekil 6.23’de TF-IDF yöntemi, DVM modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %92, nötr değeri %88 ve pozitif değeri %93 olarak tespit edilmiştir.

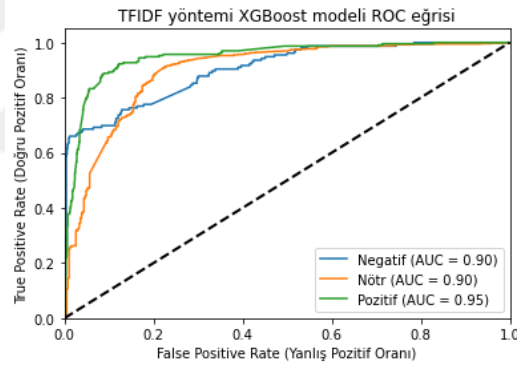
Şekil 6.24’te TF-IDF yöntemi, KA modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %82, nötr değeri %84 ve pozitif değeri %89 olarak tespit edilmiştir.

Şekil 6.25’de TF-IDF yöntemi, RO modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %87, nötr değeri %88 ve pozitif değeri %94 olarak tespit edilmiştir.

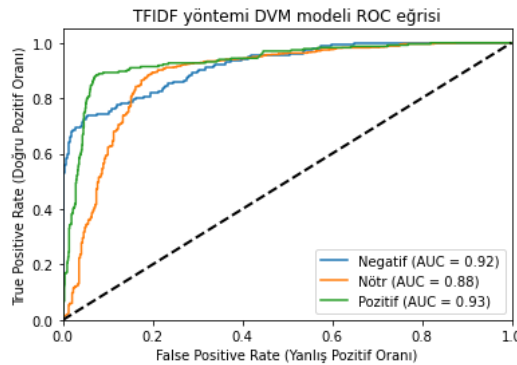
Şekil 6.26’da TF-IDF yöntemi, EYK modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %67, nötr değeri %66 ve pozitif değeri %68 olarak tespit edilmiştir.

Şekil 6.27’de TF-IDF yöntemi, GNB modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %63, nötr değeri %61 ve pozitif değeri %61 olarak tespit edilmiştir.

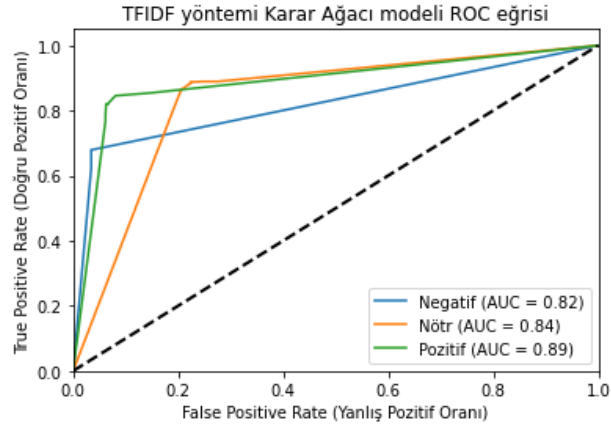
Şekil 6.28’de TF-IDF yöntemi, MNB modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %85, nötr değeri %72 ve pozitif değeri %81 olarak tespit edilmiştir.



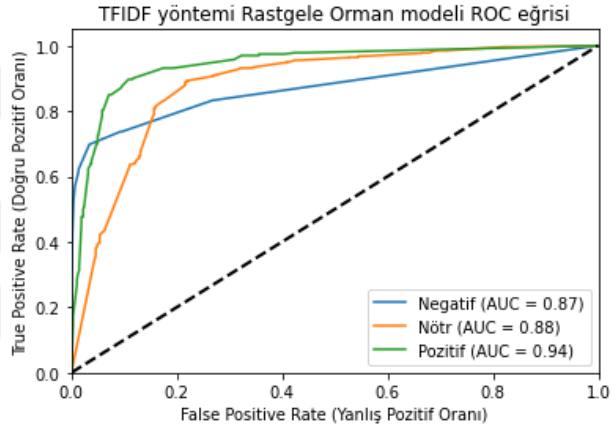
Şekil 6.22. TF-IDF yöntemi XGBoost modeli ROC analizi



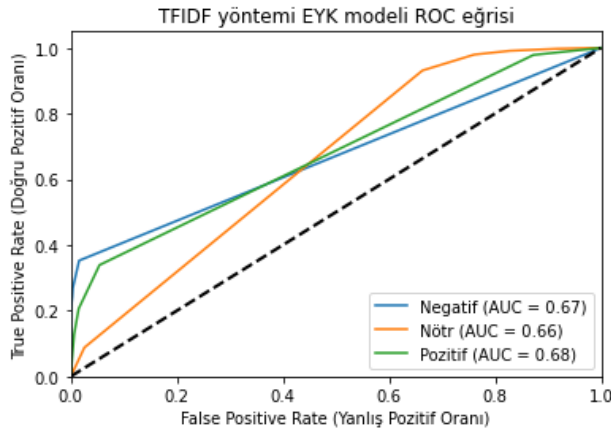
Şekil 6.23. TF-IDF yöntemi DVM modeli ROC analizi



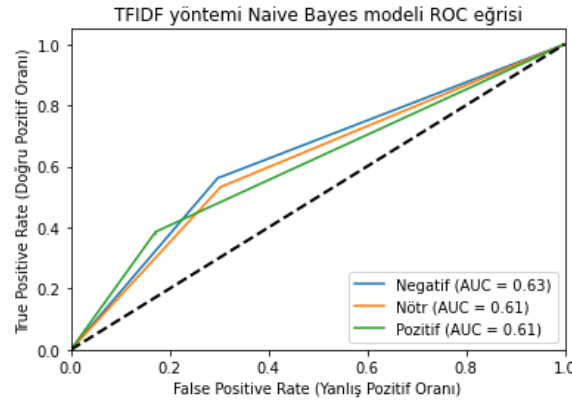
Şekil 6.24. TF-IDF yöntemi KA modeli ROC analizi



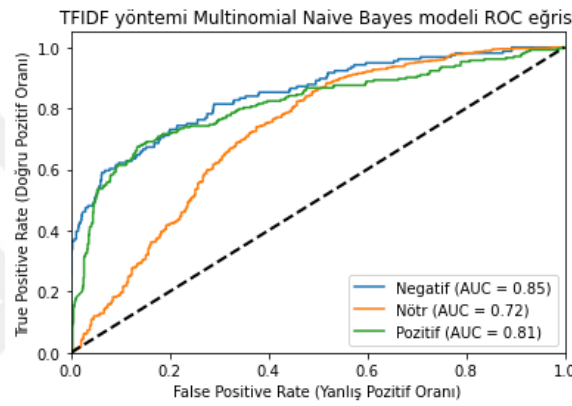
Şekil 6.25. TF-IDF yöntemi RO modeli ROC analizi



Şekil 6.26. TF-IDF yöntemi EYK modeli ROC analizi



Şekil 6.27. TF-IDF yöntemi GNB modeli ROC analizi



Şekil 6.28. TF-IDF yöntemi MNB modeli ROC analizi

Tablo 6.3'te TF-IDF yöntemine göre hiper parametre optimizasyonu sonucu en iyi parametrelerle gerçekleştirilmiş olan model başarı metrikleri yer almaktadır. EYK algoritmasında `n_neighbors` parametresi 1 seçilerek modelleme yapıldığında doğruluk, kesinlik, duyarlılık ve F1 puanı değerleri %75,6 sonucunu vermiştir. XGboost algoritmasında `max_depth` 7, `learning_rate` 0.1, `subsample` 0.5 parametreleri ile model çalıştığında doğruluk ve duyarlılık değeri %85,9 kesinlik değeri %86,3 F1 puan değeri ise %85,5 sonucunu vermiştir. RO algoritmasında `max_depth` 80, `max_features` 2, `min_samples_leaf` 1, `min_samples_split` 6, `n_estimators` 100 parametreleri ile çalıştığında doğruluk ve duyarlılık değeri %66,8 kesinlik %44,7 ve F1 puan değeri %53,5 başarı sonucunu vermiştir. DVM algoritmasında `c` 1, `gamma` 1, `kernel` linear parametreleriyle çalıştığında doğruluk ve kesinlik %82,8 kesinlik %83,4 ve F1 puanı %82,2 performans başarısı göstermiştir. XGBoost ve RO algoritmaları en iyi parametrelerle daha az başarı sonucu verirken EYK ve DVM algoritmaları daha fazla başarılı sonuç vermiştir. Daha az

başarılı sonuç vermelerinin sebebi ise veri setinin yalnızca eğitim seti kullanıldığı ve daha az bir veri seti ile çalışılması durumu olabileceği düşünülmektedir.

Tablo 6.3. En iyi parametreler kullanıldığında TF-IDF performans metrikleri

Sınıflandırma Modelleri	Performans Metrikleri (%)			
	Acc	Pre	Rec	F1-Puanı
EYK	75,6	75,6	75,6	75,6
XGBoost	85,9	86,3	85,9	85,5
RO	66,8	44,7	66,8	53,5
DVM	82,8	83,4	82,8	82,2

Tablo 6.4'te BERT, BERT+DVM, BERT+GNB, BERT+RO hibrit modellerle yapılan çalışmaların performans metrikleri gösterilmiştir. Bert modelinde doğruluk, duyarlılık ve kesinlik değeri %85,5 F1 puanı ise %85,7 sonucunu verdiği tespit edilmiştir. BERT+DVM modelinde doğruluk ve duyarlılık değeri %82,8 kesinlik değeri %83,4 F1 puanı ise %81,8 başarı sonucunu vermiştir. BERT+GNB modelinde doğruluk ve duyarlılık %71,5 kesinlik %73,8 F1 puanı ise %72,4 başarı sonucunu verdiği görülmüştür. BERT+RO modelinde doğruluk ve duyarlılık %77,4 kesinlik %78 F1 puanı ise %74,9 performans değerini vermiştir.

Tablo 6.4. BERT ile hibrit çalışan modellerin performans metrikleri

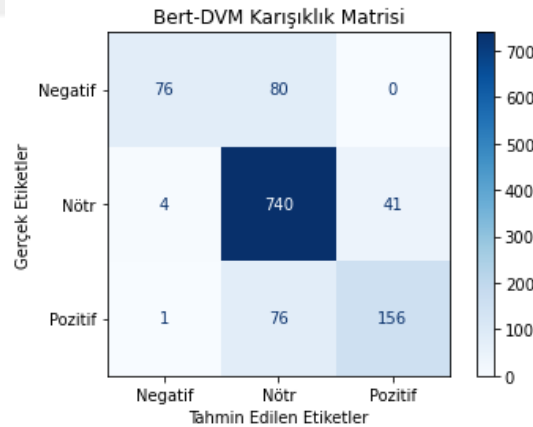
Sınıflandırma Modelleri	Performans Metrikleri (%)			
	Acc	Pre	Rec	F1-Puanı
BERT	85,8	85,8	85,8	85,7
BERT+DVM	82,8	83,4	82,8	81,8
BERT+NB	71,5	73,8	71,5	72,4
BERT+RO	77,4	78,0	77,4	74,9

Şekil 6.29'da BERT+DVM modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 76 tanesi doğru tahmin edilmiş, 80 tanesi nötr tahmin edilmiş, 0 adet ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 740 tanesi

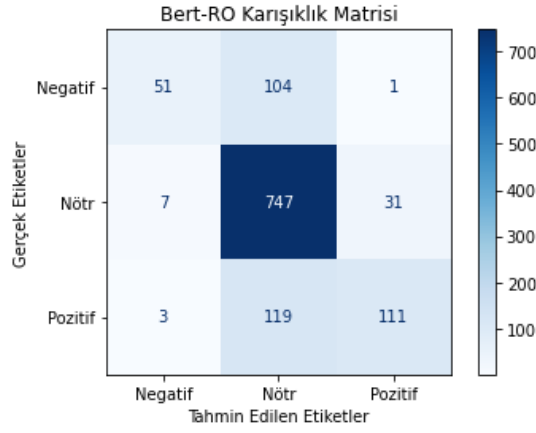
doğru tahmin edilmiş, 4 tanesi negatif tahmin edilmiş, 41 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 156 tanesi doğru tahmin edilmiş, 1 tanesi negatif, 76 tanesi nötr tahmin edilmiştir.

Şekil 6.30'da BERT+RO modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 51 tanesi doğru tahmin edilmiş, 104 tanesi nötr tahmin edilmiş, 1 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 747 tanesi doğru tahmin edilmiş, 7 tanesi negatif tahmin edilmiş, 31 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 111 tanesi doğru tahmin edilmiş, 3 tanesi negatif, 119 tanesi nötr tahmin edilmiştir.

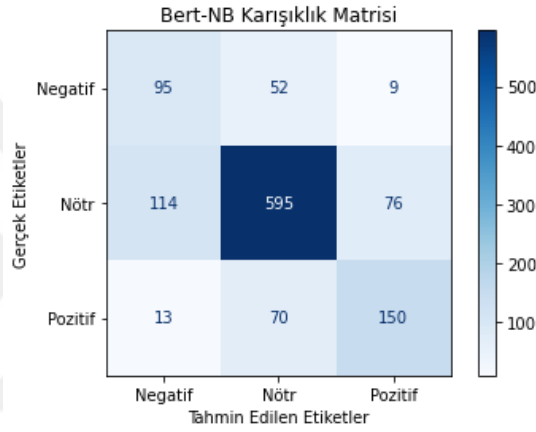
Şekil 6.31'de BERT+GNB modeli kullanılarak elde edilen karışıklık matrisi bulunmaktadır. Negatif sınıfa bakıldığında 95 tanesi doğru tahmin edilmiş, 52 tanesi nötr tahmin edilmiş, 9 tanesi ise pozitif tahmin edilmiştir. Nötr sınıfına bakıldığında 595 tanesi doğru tahmin edilmiş, 114 tanesi negatif tahmin edilmiş, 76 tanesi pozitif tahmin edilmiştir. Pozitif sınıfına bakıldığında ise 150 tanesi doğru tahmin edilmiş, 13 tanesi negatif, 70 tanesi nötr tahmin edilmiştir.



Şekil 6.29. BERT+DVM modeli karışıklık matrisi



Şekil 6.30. BERT+RO modeli karışıklık matrisi

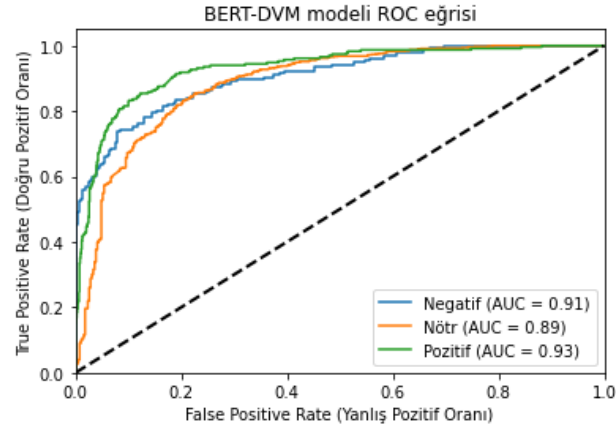


Şekil 6.31. BERT+GNB modeli karışıklık matrisi

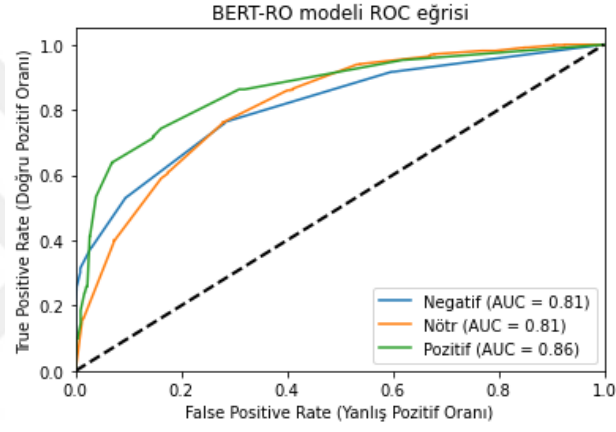
Şekil 6.32’de BERT+DVM modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %91, nötr değeri %89 ve pozitif değeri %93 olarak tespit edilmiştir.

Şekil 6.33’de BERT+RO modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %81, nötr değeri %81 ve pozitif değeri %86 olarak tespit edilmiştir.

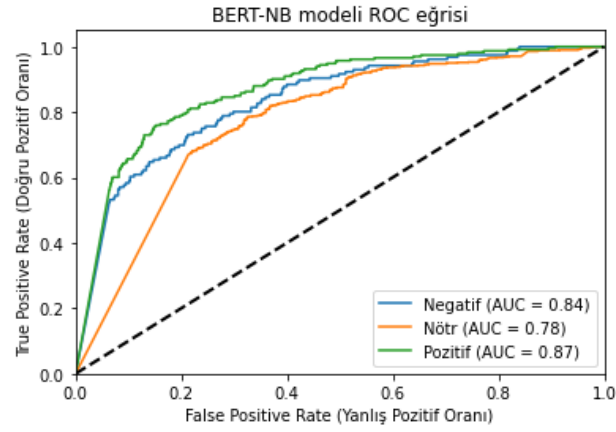
Şekil 6.34’te BERT+GNB modeli kullanılarak çizilmiş ROC eğrisi analizi bulunmaktadır. Negatif sınıflandırma için AUC değeri %84, nötr değeri %78 ve pozitif değeri %87 olarak tespit edilmiştir.



Şekil 6.32. BERT+DVM modeli ROC analizi



Şekil 6.33. BERT+RO ROC analizi



Şekil 6.34. BERT+GNB ROC analizi

7. SONUÇLAR VE ÖNERİLER

Tüm sonuçlar karşılaştırıldığında en yüksek doğruluk oranına sahip TF-IDF yöntemi kullanılarak çalıştırılan XGBoost modeli %86,4 performans sonucu olduğu tespit edilmiştir. Bu yöntem üzerinde parametre optimizasyonu uygulanıp en iyi parametrelerle çalıştığında elde edilen %85,9 performans sonucu ile tüm modeller içerisinde ikinci yüksek başarı metriğine sahip olduğu görülmüştür. Üçüncü en yüksek başarı doğruluk oranı XGboost modeli count vectorizer modeli ile %85,5 performans sonucu elde edildiği görülmüştür.

Verilere count vectorizer uygulandığında doğruluk değerinde en düşük performans değeri yüzde olarak %50,3 ve %85,5 arasında değişkenlik göstermektedir. Kesinlik değerinde %62,7 ve %83,1 arasında değişkenlik göstermektedir. Duyarlılık değerinde %50,3 ve %82,5 arasında değişkenlik göstermiştir. F1 puanında %53,8 ve %81,5 arasında değişkenlik gösterdiği gözlemlenmiştir.

Verilere TF-IDF yöntemi uygulandığında doğruluk değerinde en düşük performans değeri yüzde olarak %50,5 ve %85,5 arasında değişkenlik göstermektedir. Kesinlik değerinde %62,7 ve %83,1 arasında değişkenlik göstermektedir. Duyarlılık değerinde %50,3 ve %82,5 arasında değişkenlik göstermiştir. F1 puan değerinde %53,8 ve %81,5 arasında değişkenlik gösterdiği gözlemlenmiştir. EYK, XGBoost, RO ve DVM algoritmalarına hiper parametre optimizasyonu uygulanmıştır. Bu uygulamaların sonucunda EYK doğruluk değeri, kesinlik, duyarlılık ve F1 puan değerlerinde %75,6 verdiği gözlemlenmiştir. XGboost %85,5 ve %86,3 arasında değer verdiği gözlemlenmiştir. RO algoritmasının %53,5 ve %66,8 arasında değer verdiği gözlemlenmiştir. DVM algoritmasının ise %82,2 ve %83,4 arasında değer verdiği gözlemlenmiştir. Modellerin başarı oranları sıralandığında sırasıyla count vectorizer yönteminde XGboost, DVM, RO, KA, EYK, MNB ve GNB olarak sıralandığı gözlemlenmektedir. TF-IDF yöntemi kullanıldığında başarı oranları sıralandığında sırasıyla XGBoost, DVM, RO, KA, EYK, MNB ve GNB olarak sıralandığı gözlemlenmiştir. Her iki yöntem uygulandığında algoritmaların başarı sırasının aynı olması veri setindeki kelime dağılımlarının dengeli olmasından kaynaklandığı değerlendirilmektedir.

TF-IDF yöntemi kullanıldığında count vectorizer a göre biraz daha yüksek performans sağlamanın sebebi TF-IDF sık geçen kelimelere az ağırlık verir fakat count vectorizer her geçen kelimenin kaç kez geçtiğine baktığı için fazla ağırlık verebilir. Bu durumda fazla anlam taşımayan kelimelerin etkisini artırabilir. GNB sürekli verilerle çalışıp MNB ayrıklı verilerle çalıştığı için duygu analizinde MNB daha yüksek çıktığı gözlemlenmiştir.

EYK, XGBoost, RO ve DVM algoritmalarına hiperparametre optimizasyonu uygulanmıştır. EYK %75,6 doğruluk ve F1 puanı ile TF-IDF yöntemine göre daha iyi performans vermiştir. XGBoost %85,9 doğruluk puanı ve 85,5 F1 puanı ile TF-IDF yöntemine göre daha düşük performans vermiştir. Default değerlerde öğrenme oranı 0.3 en iyi parametrelerde 0.1 çıkmıştır ve derinlik default 6 fakat en iyi parametrede 7 çıkmıştır. Bu nedenle daha düşük bir performans sergilemiş olabilir. Düşük öğrenme oranı küçük hataları büyütüp overfittingi oluşturabilir. RO %66,8 doğruluk puanı ve 53,5 F1 puanı ile TF-IDF yöntemine göre daha düşük performans vermiştir. RO daha düşük çıkmasının sebebi eğitim verisindeki en iyi parametreler bulunmuştur. Bu parametreler test verisine uyum sağlayamamış olabileceği düşünülmüştür. DVM %82,8 doğruluk puanı ve 82,2 F1 puanı ile TF-IDF yöntemine göre daha iyi performans vermiştir

BERT modeli son yıllarda duygu sınıflandırması için en çok kullanılan yöntemlerdendir. Bu çalışmada BERT modeli makine öğrenmesi modelleriyle hibrit bir şekilde çalıştırılmış ve karşılaştırma yapılmıştır. BERT+DVM algoritması doğruluk değeri TF-IDF ve count vectorizer yöntemine göre başarı oranı daha yüksek olduğu tespit edilmiştir. BERT, dildeki bağlamı daha iyi anlaması nedeniyle ve iki iyi çalışan modelin birleşmesinden kaynaklı olarak model performansı arttığı düşünülmektedir BERT ile çalıştığında en çok başarısı artan model olan GNB algoritmasının gözle görülebilir arttığı tespit edilmiştir. TF-IDF ve count vectorizer yöntemleri ile %50 oranında olan algoritmanın doğruluğunun %71 e çıktığı gözlemlenmiştir. Bu durumun nedeni BERT modeli kelimeler arasındaki bağlamı çıkardığı için GNB algoritmasının sınıflandırmayı daha doğru ve etkin bir şekilde yapmasını sağladığı düşünülmektedir. RO algoritması ile BERT modelinin hibrit çalışması modelin performansını düşürmüştür. Bu durumun sebebi ise RO, ağaçların bağımsız olarak çalışmasını temel alır. Ancak BERT'ten gelen özellikler, genellikle birbirine bağlı derin ilişkiler içerir. Bu, ağaçların birlikte çalışmasında tutarsızlığa neden olabilir.

Tablo 7.1’de tez çalışmasıyla diğer çalışmaların karşılaştırılması yapılmıştır. Tez veri setindeki kelime dağılımı geniştir. Spesifik bir konu ele alınmamıştır. Eğer tez çalışmasında belirli bir konu üzerinde veri seti kullanılsaydı daha yüksek başarı oranına ulaşılabilirdi. Jira veri seti ile aynı veri seti kullanılmış olan çalışmada ve BERT modelinde tez çalışmasında daha düşük doğruluk oranı sergilemiştir. Kaggle platformunda yapılan çalışmada doğruluk oranı %97,9 validasyon doğruluğu ise %87 elde edilmiştir. Model ezberleme yapıyor olabilir.

Tablo 7.1. Tez ile diğer çalışmaların karşılaştırılması

Çalışma	Konu	Veri Seti	Yöntem	Başarı
Önerilen	Duygu Analizi, Jira Yorumları	Jira yorumları veri seti	BERT, XGBoost, RO,DVM , EYK, KA, GNB, MNB	Xgboost: 86,4 BERT: 85,8
Tuzcu,2020	Duygu Analizi, Kitap Yorumları	Kitap yorumu veri seti	MLP, NB, DVM, LR	MLP: %86 NB: %81 DVM: %65,71 LR: %79
Şahinaslan ve diğ.,2022	Duygu Analizi, Youtube yorumları	Youtube API veri seti	NB	NB: 65,5
Toçoğlu ve diğ., 2019	Duygu Analizi, Anket	Anket veri seti	YSA, EYK,DVM	DVM: %90,5 EYK: 56,4
Cabrera-Diego ve diğ., 2020	Duygu Analizi, Jira ve Stack Overflow yorumları	Stack Overflow ve Jira yorumları veri seti	HOMER, Rakel, Emotxt	Rakel: 58,9 HOMER: 58,5 EmoTxt: 48,8

KAYNAKLAR

- Akyol, İ. T., Özkan, N. A. Ş. (2023). Yapay zeka uygulamalarının yerel hizmet sunumuna etkisi. *Tokat Gaziosmanpaşa Üniversitesi Sosyal Bilimler Araştırmaları Dergisi*, 18(1), 120-134. DOI: 10.48145/gopsbad.1287364
- Aytaç, E. (2021). Havzaların benzerliklerini tanımlamada alternatif bir yaklaşım: Hiyerarşik kümeleme yöntemi uygulaması. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 21(4), 958-970. DOI:10.35414/akufemubid.870649
- Bilgin, M. (2017). Gerçek veri setlerinde klasik makine öğrenmesi yöntemlerinin performans analizi. *Akademik Bilişim Konferansı*, 2(9), (683–688).
- Cabrera-Diego, L. A., Bessis, N., Korkontzelos, I. (2020). Classifying emotions in Stack Overflow and JIRA using a multi-label approach. *Knowledge-Based Systems*, 195. DOI: 10.1016/j.knosys.2020.105633
- Choetkiertikul, M., Dam, H. K., Tran, T., Ghose, A. (2017). Predicting the delay of issues with due dates in software projects. *IEEE Transactions on Software Engineering*. DOI: 10.1007/s10664-016-9496-7
- Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A., Menzies, T. (2019). A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*, 45(7).
- Demir, H., Erdoğan, P., Kekeçoğlu, M. (2018). Destek vektör makineleri, YSA, K-Means ve KNN kullanarak arı türlerinin sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6(1), 47-67. DOI: 10.29130/dubited.328596
- Erçire, M., Ünsal, A. (2024). Asenkron motor eş zamanlı çoklu arızalarının titreşim sinyalleri ile çok etiketli sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 12(3), 1296-1314. DOI: 10.29130/dubited.1288799
- Eyipınar, C. D., Büyükkalkan, F., & Semiz, K. (2021). Sporcu beslenmesi ile ilgili YouTube video yorumlarının duygu analizi. *Uluslararası Beden Eğitimi Spor ve Teknolojileri Dergisi*, 2(2), 27-39.
- Gürsoy Şimşek, U., Akın, B. (2018). Sosyal medya analitiği ile değer yaratma: Duygu analizi ile geleceğe yönelim. *Makine ve İktisadi İdari Bilimler Fakültesi Dergisi*, 5(3). DOI: 10.30798/makuiibf.435804
- Heydarian, M., Doyle, T. E., Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, 10. DOI: 10.1109/ACCESS.2022.3151048
- İlhan, N., Sağaltıcı, D. (2020). Twitter’da Duygu Analizi. *Harran Üniversitesi Mühendislik Dergisi*, 5(2), 146-156. DOI: 10.46578/humder.772929

- Karaca A., Aydın, Ö. (2024). Transformatör mimarisi tabanlı derin öğrenme yöntemi ile Türkçe haber metinlerine başlık üretme. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 39(1), 485-495. DOI: 10.17341/gazimmfd.963240
- Kasongo Dahouda, M., Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9. DOI: 10.1109/ACCESS.2021.3104357
- Kaynar, O., Yıldız, M., Görmez, Y., Albayrak, A. (2016). Makine öğrenmesi yöntemleri ile duygu analizi. *International Artificial Intelligence and Data Processing Symposium (IDAP'16)*, Malatya, Türkiye, 17-18 Eylül 2016.
- Kızılkaya, Y. M., Oğuzlar, A. (2018). Bazı denetimli öğrenme algoritmalarının R programlama dili ile kıyaslanması. *Karadeniz*, (37), 90-98. DOI: 10.17498/kdeniz.405746
- Köylü, F., Atıkan, Y. (2024). Otonom araçlarda derin pekiştirmeli öğrenme yöntemleri ile sollama. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi (NOHU Journal of Engineering Sciences)*, 13(2), 429-439. DOI: 10.28948/ngumuh.1331354
- Küsmüş, A. (2024). Covid-19 Salgını Sürecinde Uzaktan Eğitime Yönelik Paylaşılan Türkçe Twitter Mesajlarının Duygu Analizi. *Mustafa Kemal Üniversitesi Eğitim Fakültesi Dergisi*, 8(13), 16-29. DOI: 10.56677/mkuefder.1467113
- Onan, A. (2022). Türkçe metin madenciliği için dikkat mekanizması tabanlı derin öğrenme mimarilerinin değerlendirilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (34), 403-407. DOI: 10.31590/ejosat.1082379
- Onan, A. (2020). Evrişimli sinir ağı mimarilerine dayalı Türkçe duygu analizi. *Avrupa Bilim ve Teknoloji Dergisi*, 374-380. DOI: 10.31590/ejosat.780609
- Ozan, Ş., Özdiş, U., Taşar, D. E., Arslan, B., & Polat, G. (2021). BERT modelinin sınıflandırma doğruluğunun sıfır-atış öğrenmesi ile artırılması. *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 14(2), 99-108. DOI: 10.54525/tbbmd.1004781
- Özkan, M., Kar, G. (2022). Türkçe dilinde yazılan bilimsel metinlerin derin öğrenme tekniği uygulanarak çoklu sınıflandırılması. *Mühendislik Bilimleri ve Tasarım Dergisi*, 10(2), 504-519. DOI: 10.21923/jesd.973181
- Özluer Başer, B., Yangın, M., Sarıdaş, E. S. (2021). Makine Öğrenmesi Teknikleriyle Diyabet Hastalığının Sınıflandırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 25(1), 112-120. DOI: 10.19113/sdufenbed.842460
- Polatgil, M. (2023). Veri ölçekleme ve eksik veri tamamlama yöntemlerinin makine öğrenmesi yöntemlerinin başarısına etkisinin incelenmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 11(1), 78-88. DOI: 10.29130/dubited.948564
- Sjarif, N. N. A., Azmi, N. F. M., Chuprat, S., Sarkan, H. M., Yahya, Y., & Sam, S. M.

- (2019). SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, 161, 509-515. DOI: 10.1016/j.procs.2019.11.150.
- Şahinaslan, Ö., Dalyan, H., Şahinaslan, E. (2022). Naive Bayes sınıflandırıcısı kullanılarak YouTube verileri üzerinden çok dilli duygu analizi. *Bilişim Teknolojileri Dergisi*, 15(2), 221-229. DOI: 10.17671/gazibtd.999960
- Toçoğlu, M. A., Çelikten, A., Aygün, İ., & Alpkoçak, A. (2019). Türkçe Metinlerde Duygu Analizi için Farklı Makine Öğrenmesi Yöntemlerinin Karşılaştırılması. *DEUFMD*, 21(63), 719-725. DOI: 10.21205/deufmd.2019216303
- Tomak, L., Bek, Y. (2010). İşlem karakteristik eğrisi analizi ve eğri altında kalan alanların karşılaştırılması. *Journal of Experimental and Clinical Medicine*, 27(2).
- Tosunoğlu, E., Yılmaz, R., Özeren, E., Sağlam, Z. (2021). Eğitimde makine öğrenmesi: Araştırmalardaki güncel eğilimler üzerine inceleme. *Ahmet Keleşoğlu Eğitim Fakültesi Dergisi*, 3(2), 178-199. DOI: 10.38151/akef.2021.16
- Tuzcu, S. (2020). Çevrimiçi kullanıcı yorumlarının duygu analizi ile sınıflandırılması. *Estudam Bilişim Dergisi*, 1(2), 1-5.
- URL-1: <https://www.atlassian.com/software/jira>, (Ziyaret tarihi: 28 Şubat 2025).
- URL-2: <https://pandas.pydata.org/>, (Ziyaret tarihi: 15 Ocak 2025).
- URL-3: <https://scikit-learn.org/stable/>, (Ziyaret tarihi: 16 Ocak 2025).
- URL-4: Mongelli, L. (2011). Kaggle. <https://www.kaggle.com/code/lucamongelli/jira-wbert/>, (Ziyaret tarihi: 20 Ocak 2025).
- Yaman, O., Tekin, R. (2023). Akıllı ev sistemleri için XGBoost tabanlı saldırı tespit yöntemi. *Zeki Sistemler Teori ve Uygulamaları Dergisi*, 6(2), 152-158. DOI: 10.38016/jista.1075054
- Yürütücü, Ö. Y., Demir, Ş. (2023). Ön eğitilmiş dil modelleriyle duygu analizi. *İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 5(1), 46-53. DOI: 10.47769/izufbed.1312032.
- Zilyas, D., Yılmaz, A. (2023). Makine öğrenmesi yöntemleri ile eğitim başarısının tahmini modeli. *DUJE (Dicle University Journal of Engineering)*, 14(3), 437-447. DOI: 10.24012/dumf.1322273

KİŞİSEL YAYIN VE ESERLER

Emen, M., Konyar, M. Z. (2024). Proje Takip Yazılımı Jira’da Yapay Zeka Tabanlı Duygu Analizi, *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, Türkiye, 21-22 Eylül 2024.



ÖZGEÇMİŞ

2019 yılında Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünden mezun olmuştur. 2020 yılında 2 yıl 8 ay özel bir şirkette yazılım geliştirme mühendisi olarak görev almıştır. 2022 yılında Kocaeli Üniversitesi Yazılım Mühendisliği bölümünde yüksek lisans eğitimine başlamış ve devam etmektedir. 2022 yılından beri özel bir bankada kıdemli uzman takım lideri olarak çalışma hayatına devam etmektedir.

