



T.C.
SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

**Çok Kanallı Parfüm İncelemeleri için Derin Öğrenme Tabanlı Analiz ve
Öneri Sistemi**

Ömerhan SEZGİN
MÜHENDİSLİK TASARIMI

Mayıs-2025
KONYA
Her Hakkı Saklıdır

PROJE KABUL VE ONAYI

Ömerhan Sezgin tarafından hazırlanan “Çok Kanallı Parfüm İncelemeleri için Derin Öğrenme Tabanlı Analiz ve Öneri Sistemi” adlı proje çalışması 26/05/2025 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği/oy çokluğu ile Selçuk Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği bölümünde Mühendislik Tasarımı Projesi olarak kabul edilmiştir.

Jüri Üyeleri

Danışman

Dr. Öğr. Üyesi ALİ ÇETİNKAYA

İmza

Yukarıdaki sonucu onaylarım.

Bilgisayar Mühendisliği
Bölüm Başkanı

PROJE BİLDİRİMİ

Bu projedeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve proje yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by project rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ömerhan SEZGİN

Tarih: 26/05/2025

ÖZET

MÜHENDİSLİK TASARIMI

Çok Kanallı Parfüm İncelemeleri için Derin Öğrenme Tabanlı Analiz ve Öneri Sistemi

Ömerhan SEZGİN

SELÇUK ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Danışman: Dr. Öğr. Üyesi ALİ ÇETİNKAYA

2025, 66 Sayfa

Jüri

Dr. Öğr. Üyesi ALİ ÇETİNKAYA

Bu çalışmada, çevrimiçi parfüm incelemelerini çok yönlü bir şekilde analiz eden ve kişiselleştirilmiş öneriler sunan derin öğrenme tabanlı bir sistem geliştirilmiştir. Sistem; metin, sayısal ve kategorik verileri bir araya getirerek çok kanallı bir yapıyla parfüm değerlendirmeleri yapmaktadır. Doğal Dil İşleme (NLP) teknikleri ile kullanıcı yorumlarındaki duygusal ve yapısal bilgiler analiz edilmekte, RoBERTa modeli ile metinler kodlanmakta, parfüm notaları ise TF-IDF vektörizasyonu ve kosinüs benzerliği kullanılarak tespit edilmektedir. Sistem aynı zamanda parfümlerin kalıcılık, yayılım ve cinsiyet gibi özelliklerini de tahmin edebilmekte ve bu çıktılara göre kullanıcıya önerilerde bulunmaktadır. Elde edilen sonuçlar, metin tabanlı incelemelerin ve derin öğrenme algoritmalarının parfüm tercihlerine dair içgörüler üretmede etkili olduğunu göstermektedir.

Anahtar Kelimeler: Çok Kanallı Mimari, Derin Öğrenme, Doğal Dil İşleme, Kosinüs Benzerliği, Öneri Sistemi, Parfüm İncelemesi, RoBERTa, TF-IDF.

ABSTRACT

ENGINEERING DESIGN

Deep Learning-Based Analysis and Recommendation System for Multi-Channel Perfume Reviews

Ömerhan SEZGİN

**SELCUK UNIVERSITY
FACULTY OF TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING**

Advisor: Asst. Prof. Dr. Ali Çetinkaya

2025, 66 Pages

**Jury
Asst. Prof. Dr. Ali Çetinkaya**

In this study, a deep learning-based system was developed to analyze online perfume reviews and provide personalized recommendations. The system evaluates perfumes through a multi-channel architecture by integrating textual, numerical, and categorical data. Sentimental and structural insights in user reviews are extracted using Natural Language Processing (NLP) techniques, while texts are encoded with the RoBERTa model and perfume notes are identified using TF-IDF vectorization and cosine similarity. Additionally, the system predicts characteristics such as longevity, sillage, and gender preference of the perfumes, providing tailored suggestions accordingly. Results indicate that text-based reviews combined with deep learning algorithms are effective in generating insights into perfume preferences.

Keywords: Cosine Similarity, Deep Learning, Multi-Channel Architecture, Natural Language Processing, Perfume Review, Recommendation System, RoBERTa, TF-IDF.

ÖNSÖZ

Bu proje çalışması, günümüzde dijital platformlarda hızla artan kullanıcı yorumlarının anlamlı şekilde analiz edilerek değerli bilgiler elde edilmesini amaçlamaktadır. Özellikle parfüm endüstrisi gibi duyuşal ve subjektif değerlendirmelerin ön planda olduđu alanlarda, geleneksel yaklaşımların yetersiz kaldığı durumlarda yapay zekâ ve derin öğrenme yöntemlerinin katkısı oldukça büyüktür. Bu bağlamda geliştirilen “Çok Kanallı Parfüm İncelemeleri için Derin Öğrenme Tabanlı Analiz ve Öneri Sistemi” adlı bu çalışma; metin, sayısal ve kategorik verileri bütüncül bir yapıda işleyerek parfüm önerileri sunmayı hedeflemiştir.

Çalışma sürecinde akademik kaynaklara dayalı teorik bilgi ile uygulama geliştirme aşamalarının entegrasyonu hem araştırma becerilerimin hem de yazılım yetkinliğimin gelişmesini sağlamıştır. Bu sürecin her aşamasında bana rehberlik eden, bilgi ve tecrübesiyle yol gösteren değerli danışmanım Dr. Öğr. Üyesi Ali Çetinkaya'ya en içten teşekkürlerimi sunarım.

Ayrıca, araştırma sürecinde desteklerini esirgemeyen aileme, motivasyon kaynağım olan arkadaşlarıma ve çalışmamı anlamlı kılacak fikir alışverişlerinde bulunduğum tüm paydaşlara şükranlarımı sunarım.

Bu çalışmanın hem akademik camiaya hem de parfüm endüstrisinde yapay zekâ temelli sistemlerin gelişimine katkı sağlamasını temenni ederim.

Ömerhan SEZGİN

Konya / 2025

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	iv
1. GİRİŞ	1
1.1.Parfüm Endüstrisinde Veri Analizinin Önemi	1
1.1.1.Verinin Gücü: Parfüm Pazarını Anlamak:.....	2
1.2.Parfüm İncelemelerinin Karmaşık Yapısı.....	2
1.3.Yapay Zeka ve Parfüm Endüstrisi Kesişimi	3
2. KAYNAK ARAŞTIRMASI	5
2.1. Parfüm İncelemelerinde Metin Madenciliği Çalışmaları.....	5
2.2. Çok Modlu Veri Analizi Yaklaşımları.....	6
2.3. Parfüm Bileşenlerinin Yapısal Analizi	6
2.4. Öneri Sistemlerinde Derin Öğrenme Uygulamaları	7
3. MATERYAL VE YÖNTEM.....	9
3.1. Veri Yapısı ve Önışleme	9
3.1.1. Not Sözlüğü Oluşturma ve Yönetimi.....	9
3.1.2. Metin İşleme ve Nota Tespiti.....	13
3.1.3. Tokenizasyon ve Metin Önışleme	19
3.2. Model Mimarisi ve Çok Kanallı Yapı	21
3.2.1. RoBERTa Tabanlı Metin Kodlama: Pre-trained avantajları, havuzlama stratejisinin etkileri	24
3.2.2. Sayısal ve Kategorik Özellikler	26
3.2.3. Özellik Birleştirmeye ve Dikkat Mekanizması	29
3.3 Model Girdileri ve Çıktıları	31
3.3.1. Metin ve Özellik Girişleri	31
3.4. Parfüm Analiz Sistemi	37
3.4.1. PerfumeReviewAnalyzer Sınıfı	38
3.4.2. Özellik İşleme ve Yorumlama	39
3.4.3. Nota Tespiti ve Özellik Hazırlama	40
3.4.4. Tahmin Sonuçlarının Değerlendirilmesi.....	41
3.5. Parfüm Tarifi ve Özellik Analizi	43
3.5.1. Koku Özellikleri Dönüşümü.....	43
3.5.2. Tarif Oluşturma Mekanizması	44
3.6. Cinsiyet Analizi ve Tahmin Birleşimi	46
3.6.1. Ağırlıklandırılmış Puanlama Yaklaşımı	47

3.6.2. Skorlama ve Varsayılan Değerler (0-5 Cinsiyet Ölçeği)	47
3.6.3. Varsayılan Değer ve Unisex Etkisi	47
3.6.4. Model + Metin Tahmin Birleşimi	48
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	49
4.1. Test Verileri ve Özellikleri	49
4.2. Verilerin Test Sonuçları (Uygulama Çıktıları)	50
4.3.Reel Çıktı Analizi	57
4.4.Tartışma	58
5. SONUÇLAR VE ÖNERİLER.....	60
5.1. Sonuçlar	60
5.2. Öneriler	61
KAYNAKLAR	63
EKLER	66

SİMGELER VE KISALTMALAR

Kısaltmalar

- ABD: Amerika Birleşik Devletleri
- CAGR: (En: Compound Annual Growth Rate) Yıllık bileşik büyüme oranı
- CSV: (En: Comma-Separated Values) Virgülle ayrılmış değerler
- NLP: (En: Neuro Linguistic Programming) Sinir dili programlaması
- YZ: Yapay zeka

1. GİRİŞ

Parfüm pazarı, küresel ölçekte önemli bir büyüklüğe ulaşmış ve sürekli olarak büyüme göstermektedir. 2022 yılında küresel parfüm pazarının 50,85 milyar ABD doları değerinde olduğu tahmin edilmektedir ve 2030 yılına kadar %5,9'luk bir bileşik yıllık büyüme oranı (CAGR) ile büyümesi beklenmektedir. (Grand View Research, 2011 - 2021) Bir başka tahmine göre ise, pazarın 2022'de 40.13 milyar ABD doları büyüklüğünde olduğu ve 2030'a kadar yaklaşık %6,07'lik bir CAGR ile 64,41 milyar ABD dolarına ulaşacağı öngörülmektedir. (Zion Market Research, 2022) Bu büyüme oranları, kişisel bakıma verilen önemin artması ve lüks ile egzotik kokulara olan talebin yükselmesi gibi faktörlerden kaynaklanmaktadır. Farklı kaynaklardan elde edilen pazar büyüklüğü verilerindeki bu küçük farklılıklar, kullanılan farklı metodolojilerden veya pazar tanımlarından kaynaklanabilir. Bu nedenle, pazar büyüklüğü tahminlerine temkinli yaklaşmak ve farklı kaynakları dikkate almak önemlidir.

Yoğun rekabetin yaşandığı bu sektörde, parfüm şirketlerinin pazarda öne çıkabilmeleri için veri analizinin rolü giderek artmaktadır. Tüketici tercihlerinin ve pazar trendlerinin hızlı bir şekilde değişmesi (Grand View Research, 2011 - 2021), şirketlerin rekabette kalabilmeleri için bu değişimlere hızla adapte olmalarını gerektirmektedir. Veri analizi, şirketlere tüketici davranışlarını anlama ve iş stratejilerini buna göre yönlendirme konusunda önemli bir rekabet avantajı sağlamaktadır. (Faster Capital, 2025) Bu durum, parfüm endüstrisinde geleneksel sezgisel kararlardan veri destekli yaklaşımlara doğru bir kaymaya neden olmaktadır

1.1. Parfüm Endüstrisinde Veri Analizinin Önemi

Küresel parfüm endüstrisi, sürekli büyüme ve yoğun rekabetin karakterize ettiği dinamik bir ortamdır. Tüketici davranışlarını anlamak ve iş stratejilerini yönlendirmek için veri analizinin rolü giderek daha kritik hale gelmektedir. Bu rapor, parfüm endüstrisindeki veri analizinin önemini ve temel uygulamalarını incelemektedir. Veri analizinin, tüketici tercihlerini anlamaktan, pazar trendlerini belirlemeye, ürün geliştirmeyi optimize etmekten, pazarlama stratejilerini iyileştirmeye ve operasyonel verimliliği artırmaya kadar birçok alanda parfüm endüstrisine önemli faydalar sağladığı görülmektedir. Raporun temel bulguları, veri odaklı yaklaşımların parfüm şirketlerinin rekabet avantajı elde etmeleri ve sürdürülebilir büyüme sağlamaları için vazgeçilmez olduğunu göstermektedir.

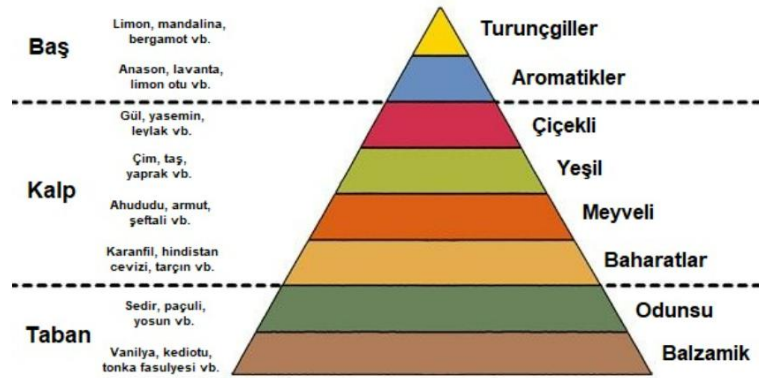
1.1.1. Verinin Gücü: Parfüm Pazarını Anlamak:

Küresel parfüm pazarının büyüklüğü ve büyüme trendleri, endüstri oyuncuları için önemli bilgiler sunmaktadır. 2022 yılında küresel parfüm pazarının 40,13 milyar ABD doları ile 50,85 milyar ABD doları arasında bir değere sahip olduğu tahmin edilmektedir. (Grand View Research, 2011 - 2021) Geleceğe yönelik tahminler ise pazarın önemli ölçüde büyüyeceğini göstermektedir. 2030 yılına kadar pazarın 64,41 milyar ABD doları ile 80,16 milyar ABD doları arasına ulaşması beklenirken, 2034 yılına yönelik bir projeksiyon ise pazar büyüklüğünün 101,47 milyar ABD dolarına kadar çıkabileceğini öngörmektedir. (Precedence Research, 2025) Bu büyüme, 2023'ten 2030'a kadar %5,9 ile %6,07 arasında ve 2025'ten 2034'e kadar %5,88 olarak tahmin edilen yıllık bileşik büyüme oranları (CAGR) ile desteklenmektedir. Bu büyümenin temelinde, kişisel bakıma artan ilgi, lüks ve egzotik kokulara olan talep ve tüketicilerin artan harcanabilir gelirleri yatmaktadır. Farklı araştırma raporlarındaki pazar büyüklüğü ve CAGR rakamları genel bir büyüme trendini işaret etmekle birlikte, kesin rakamlar metodoloji ve kapsam farklılıkları nedeniyle değişiklik gösterebilir. Bu durum, endüstri analizlerinde birden fazla kaynağa başvurmanın ve sonuçları bağlam içinde değerlendirmenin önemini vurgular.

Geçmişte parfüm endüstrisi büyük ölçüde deneyimli parfümörlerin sezgisel kararlarına dayanırken, günümüzde rekabet avantajı elde etmek ve daha bilinçli kararlar almak için veri analizine giderek daha fazla güvenilmektedir. Geleneksel sezgisel kararlardan veri destekli yaklaşımlara geçiş, endüstride önemli bir değişim yaratmaktadır. Veri analizi, riskleri azaltma ve verimliliği artırma potansiyeli sunarak (Mia Setya Rahayu, 2024), parfüm şirketlerinin stratejik hedeflerine ulaşmalarına yardımcı olmaktadır.

1.2. Parfüm İncelemelerinin Karmaşık Yapısı

Parfüm incelemelerinin yapısı genellikle parfümün farklı aşamalarını tanımlar: üst, orta ve alt notalar. Üst notalar, parfümün ilk uygulandığında hissedilen ilk izlenimdir ve genellikle hafif ve uçucudur. Orta notalar, üst notalar kaybolduktan sonra ortaya çıkar ve parfümün kalbini ve karakterini oluşturur. Alt notalar ise parfümün temelini oluşturur ve kalıcılığını sağlar. (Odosbeauty, 2025) Parfüm piramidi (Şekil 1.2.1), bu notaların etkileşimini görselleştiren bir modeldir. Bu yapı, tüketicilerin parfümün genel karakterini ve zaman içindeki evrimini anlamalarına yardımcı olur.



Şekil 1.2.1 Parfüm piramidi

Metin tabanlı parfüm incelemeleri, kokuları tanımlamada kullanılan dilin çeşitliliği ve yaratıcılığı sayesinde subjektif deneyimleri yakalamada zengin ve derinlemesine bilgiler sunar. Bu incelemeler, kokunun sadece notalarını değil, aynı zamanda kullanıcının duygusal tepkilerini, kişisel deneyimlerini ve parfümle kurduğu bağlantıları da ifade etme yeteneğine sahiptir. Doğal Dil İşleme (NLP) teknikleri, bu metin tabanlı incelemelerden anlamlı içgörüler elde etmek için güçlü bir potansiyel sunar. (Faster Capital, 2025)

Parfümleri değerlendirmede sayısal derecelendirmeler de rol oynar ve genellikle popülerlik ve genel beğeni göstergesi olarak kullanılır. Ancak, sayısal derecelendirmelerin bağlam ve gerekçe eksikliği nedeniyle sınırlamaları vardır. Ayrıca, bireysel tercihlerdeki farklılıkların sayısal puanlara yansıtılması da zordur. Bu nedenle, parfümleri değerlendirmede en etkili yaklaşım, sayısal derecelendirmelerin ve metin tabanlı subjektif değerlendirmelerin birleştirilmesidir. Sayısal puanlar genel bir bakış sağlarken, metin incelemeleri kokunun nüanslarını ve kullanıcının kişisel deneyimini anlamak için ayrıntılı bilgiler sunar. Duygusal tepkileri ve kişisel tercihleri anlamak için metin analizinin önemi büyüktür. (Larisa Nikitina, 2023)

1.3. Yapay Zeka ve Parfüm Endüstrisi Kesişimi

Yapay zeka (YZ), koku yaratma, öneri ve pazarlamada parfüm endüstrisinde giderek artan bir rol oynamaktadır. YZ algoritmaları, büyük veri kümelerini analiz ederek yeni koku kombinasyonları önerebilir. Tüketici tercihlerine göre kişiselleştirilmiş parfüm önerileri sunulabilir. Pazarlama materyalleri ve görsel içerik oluşturmada YZ'den yararlanılabilir. Ayrıca, YZ verimli ürün geliştirme süreçlerine katkıda bulunabilir.

Ancak, yapay zekanın, özellikle parfümerinin sanatsal ve subjektif yönlerini yakalamadaki zorlukları ve sınırlamaları da bulunmaktadır. İnsan duygularını ve

yaratıcılığını tam olarak taklit etmekte zorlanabilir. Koku algısının öznel ve kişisel doğası, YZ'nin bu alandaki etkinliğini sınırlayabilir. Yapay zekâ tarafından üretilen kokularda özgünlük ve derinlik eksikliği potansiyeli mevcuttur. Parfüm tasarımı, özünde sanatsal bir süreçtir ve YZ'nin bu sanatsal ve subjektif yönleri tam olarak yakalaması hala zor gözükmetedir.

Parfüm tasarımında insan yaratıcılığının yapay zeka karşısındaki rolü devam etmektedir bu nedenle Yapay zeka, parfümörlerin yaratıcılığını artırabileceği bir araç olarak kullanılabilir. Son kararın ve sanatsal dokunuşun her zaman parfümörlere ait olması beklenir. Yapay zekanın rutin görevleri otomatikleştirmesiyle parfümörler, yaratıcı çalışmalara daha fazla odaklanabilirler. Yapay zeka, parfüm tasarım sürecinde parfümörlere yardımcı olabilecek güçlü bir araç olsa da insan yaratıcılığının ve sanatsal yerini alması beklenmemektedir.

2. KAYNAK ARAŞTIRMASI

Parfüm incelemeleri üzerine yapılan metin madenciliği çalışmaları, proje yürütücüleri olarak bizce tüketicilerin parfümlerle ilgili görüş ve deneyimlerini derinlemesine anlamak için oldukça değerli. Bu tür çalışmalar, doğal dil işleme teknikleri kullanarak geniş çaplı metin verisini analiz ediyor ve böylece tüketicilerin parfüm hakkındaki duygusal tutumlarını ortaya koyabiliyor. Özellikle duygu analizi, bir parfümün genel algısının olumlu mu yoksa olumsuz mu olduğunu anlamada çok faydalı oluyor. Konu modellemesi ise parfüm incelemelerinde sıkça tekrarlanan temaları ve parfüm notalarını tespit etmeye olanak sağlıyor.

2.1. Parfüm İncelemelerinde Metin Madenciliği Çalışmaları

Nikitina ve Furuoka'nın (2024) çalışması (Larisa Nikitina, 2023), bu alandaki önemli örneklerden biri. Çevrimiçi parfüm incelemelerinin dilbilimsel analizini yaparak olumlu ve olumsuz incelemeler arasındaki tematik ve kelime kullanımı farklılıklarını ortaya koymuşlar. Bu çalışmanın bulguları, olumlu yorumların sosyal dünya, sosyal süreçler, cinsellik, boş zaman ve zihinsel durumlarla ilgili kelimeleri daha fazla içerdiğini gösteriyor. Buna karşılık, olumsuz incelemelerde sağlık, hastalık ve sosyal çatışma ile ilgili kavramların daha sık geçtiği dikkat çekici. Ayrıca konu modellemesi analizine göre, olumlu yorumlarda parfüm kullanım deneyimleri, duyuşsal karakter ve koku kaliteleri ön plana çıkarken, olumsuz yorumlarda cilt kimyası, parfüm notaları ve fiziksel reaksiyonlar üzerinde durulduğu görülüyor.

Bu tarz metin madenciliği çalışmaları, parfüm firmalarının tüketici geri bildirimlerini daha iyi kavrayıp, ürün geliştirme ve pazarlama stratejilerini daha etkin hale getirmesine önemli katkılar sunuyor. Böylelikle müşteri memnuniyetinin artması da mümkün oluyor.

2.2. Çok Modlu Veri Analizi Yaklaşımları

Çok modlu veri analizi, parfüm endüstrisinde tüketici algısını ve tercihlerini anlamak için giderek daha önemli bir yaklaşım haline gelmektedir. Bu yaklaşım, metin, görüntü ve hatta biyometrik veriler gibi farklı veri türlerini birleştirerek parfüm deneyiminin daha kapsamlı bir resmini sunar. (Owkin, tarih yok) Örneğin, parfüm reklamlarının analizi, görsel ve işitsel unsurların koku algısını nasıl etkilediğini ortaya koyabilir.

MediaEval 2023 tarafından düzenlenen Musti görevi, metin ve görüntülerdeki koku referanslarını tanımak ve bu referansları farklı modaliteler arasında bağlamak için çok dilli ve çok modlu teknolojilerin geliştirilmesini amaçlamaktadır. Bu tür çalışmalar, kültürel miras koleksiyonlarının anlaşılmasını zenginleştirmek ve müzeler ile galeriler için daha sürükleyici ve kapsayıcı deneyimler yaratmak için potansiyel sunmaktadır. (MediaEval, 2023)

Ayrıca, duygusal veya görsel girdileri bir parfüm kataloğundaki en uygun ürünle eşleştirmek için multimodal yapay zeka kullanan araçlar hem çevrimiçi hem de mağaza içi müşterilere daha kişiselleştirilmiş öneriler sunarak perakendecilere ve markalara fayda sağlayabilir. Bu tür yaklaşımlar, tüketicilerin koku dilini bilmelerine gerek kalmadan ruh hallerine, anılarına veya beğenilerine göre parfüm keşfetmelerine olanak tanır.

2.3. Parfüm Bileşenlerinin Yapısal Analizi

Parfümün yapısal analizi, bir kokuyu oluşturan farklı bileşenlerin (notalar) ve bu bileşenlerin parfümün genel karakteri üzerindeki etkisinin incelenmesini içerir. Parfüm notaları genellikle üç katmana ayrılır: parfümün ilk izlenimini veren üst notalar, parfümün kalbini oluşturan orta notalar ve parfümün temelini ve kalıcılığını sağlayan alt notalar. (The olfactive pyramid, tarih yok)

Koku çarkı (Şekil 2.3.1), parfüm notalarını ve koku ailelerini (çiçeksi, oryantal, odunsu, ferahlatıcı) anlamak için kullanılan görsel bir araçtır. Bu çark, farklı kokular arasındaki ilişkileri ve benzerlikleri göstererek tüketicilere ve parfüm uzmanlarına yol gösterir.



Şekil 2.3.1 Koku Çarkı

Kimyasal analiz teknikleri, özellikle gaz kromatografisi-kütle spektrometrisi (GC-MS), parfüm bileşenlerini ayırmak, tanımlamak ve ölçmek için ideal yöntemlerdir. Bu analizler, parfüm formülasyonlarının kalitesini ve doğruluğunu kontrol etmek için önemlidir. (Contract Laboratory, tarih yok)

2.4. Öneri Sistemlerinde Derin Öğrenme Uygulamaları

Projede yer alan derin öğrenme, parfüm endüstrisindeki öneri sistemlerinde giderek daha fazla kullanılmaktadır. Bu sistemler, kullanıcıların geçmiş tercihlerini, parfüm notalarını ve diğer özellikleri analiz ederek kişiselleştirilmiş parfüm önerileri sunar.

Örneğin, NLP tabanlı derin öğrenme yaklaşımları, parfüm açıklamalarını ve notalarını anlamak ve benzer parfümleri önermek için kullanılacaktır. Bu modeller, parfüm notalarını semantik olarak anlamlı vektörlere dönüştürerek benzerlik ölçütleri aracılığıyla karşılaştırma yapmayı mümkün kılacağı düşünülerek proje kapsamında kullanılmıştır.

Ayrıca, tüketici geri bildirimlerini ve derecelendirmelerini kullanarak parfüm tercihlerini tahmin etmek için derin sinir ağıları da geliştirilmiştir. Bu tür sistemler, parfüm perakendecilerinin çevrimiçi satışlarını artırmalarına ve müşteri memnuniyetini iyileştirmelerine yardımcı olabilir.

3. MATERYAL VE YÖNTEM

Bu çalışmada, metin dokümanları arasındaki semantik benzerlikleri tespit etmek için TF-IDF vektörizasyonu ve kosinüs benzerliği ölçütleri kullanılmıştır. Önerilen yöntem, dokümanlardaki tekli ve ikili kelime gruplarını yakalayarak anlamsal bütünlüğü korurken, benzerlik eşik değeri ile alakalı içeriklerin filtrelenmesini sağlamaktadır.

3.1. Veri Yapısı ve Önışleme

Veriler Fragrantica (Fragrantica, tarih yok) adlı siteden Python temelli kodlar ve kütüphaneler, Web Scraping metotlarına uygun bir şekilde kullanılarak elde edilmiş olup alt başlıklara bölünerek koku notaları için “NoteDictionary” başlığı adı altında bir sözlük yapısı oluşturulmuştur.

3.1.1. Not Sözlüğü Oluşturma ve Yönetimi

Not sözlüğü için veriler Python temelli “Selenium” kütüphanesi kullanılarak site için belirtilen yollar üzerinden gerekli veriler “csv” dosya formatına aktararak işlem gerçekleştirilmiştir. (Şekil 3.1.1.1.)

```

anadenemeler - webscraping.py
1 import csv
2 from selenium import webdriver
3 from selenium.webdriver.chrome.service import Service
4 from webdriver_manager.chrome import ChromeDriverManager
5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.support.ui import WebDriverWait
7 from selenium.webdriver.support import expected_conditions as EC
8
9 service = Service(ChromeDriverManager().install())
10 driver = webdriver.Chrome(service=service)
11
12 driver.get("https://www.fragrantica.com/notes/")
13
14 WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.XPATH, '//*[@id="groupnotes_group_01_title"]/div/h2')))
15
16 kategori_xpath_list = [
17     '//*[@id="groupnotes_group_01_title"]/div/h2',
18     '//*[@id="groupnotes_group_02_title"]/div/h2',
19     '//*[@id="groupnotes_group_03_title"]/div/h2',
20     '//*[@id="groupnotes_group_04_title"]/div/h2',
21     '//*[@id="groupnotes_group_05_title"]/div/h2',
22     '//*[@id="groupnotes_group_06_title"]/div/h2',
23     '//*[@id="groupnotes_group_07_title"]/div/h2',
24     '//*[@id="groupnotes_group_08_title"]/div/h2',
25     '//*[@id="groupnotes_group_09_title"]/div/h2',
26     '//*[@id="groupnotes_group_10_title"]/div/h2',
27     '//*[@id="groupnotes_group_11_title"]/div/h2',
28     '//*[@id="groupnotes_group_12_title"]/div/h2',
29     '//*[@id="groupnotes_group_13_title"]/div/h2',
30 ]
31
32 with open('kategori_bilgileri.csv', mode='w', newline='', encoding='utf-8') as file:
33     writer = csv.writer(file)
34     writer.writerow(['Kategori', 'Bağlantı Metni'])
35
36 kategoriler = []
37
38 for kategori_xpath in kategori_xpath_list:
39     try:
40         kategori_element = driver.find_element(By.XPATH, kategori_xpath)
41         kategori = kategori_element.text
42         kategoriler.append(kategori)
43
44         category_section_xpath = f"//*[@id='main-content']/div[1]/div[1]/div/div[{kategori_xpath_list.index(kategori_xpath) + 8}]"
45         WebDriverWait(driver, 10).until(EC.presence_of_all_elements_located((By.XPATH, category_section_xpath)))
46
47         links = driver.find_elements(By.XPATH, f"({category_section_xpath})/div/div")
48
49         if links:
50             for link in links:
51                 text = link.text
52                 writer.writerow([kategori, text])
53         else:
54             writer.writerow([kategori, "Bağlantı Bulunamadı"])
55     except Exception as e:
56         writer.writerow([kategori_xpath, f"Hata: {e}"])
57
58 driver.quit()
59
60 print("Veriler CSV dosyasına kaydedildi.")
61

```

Şekil 3.1.1.1. Veri eldesi için yazılan program

3.1.1.1. Not sözlüğünün yapısı ve içeriği

Not sözlüğü toplamda 600'e yakın veriden oluşmaktadır bu verilerin hepsi belirtilen web sitesinden elde edilmiş olup yapılan çalışma ve araştırmalara göre daha önce bahsi geçen nota sınıflandırmasına sokularak alt, orta ve üst olmak üzer 3 nota yapısına sitede belirtilen kategori sınıflandırmasına uyularak kokular indekslenerek "default" bir sözlük oluşturulmuştur. (Şekil 3.1.2.1)

```

1  def __init__(self, vocab_file=None):
2      self.notes = []
3      self.note_to_idx = {}
4      self.idx_to_note = {}
5
6
7      default_notes = [
8
9          "bergamot", "lemon", ...

```

Şekil 3.1.1.1.1. "default_notes" Adlı sözlük yapısı

Sistemde veriye yani csv uzantılı dosyaya erişilmemesi durumunda kullanılacak olan bu sistemde ek olarak model için kokular nota ve kategorilerine göre sistemde ayrı bir şekilde tanımlanmıştır. (Şekil 3.1.2.2)

```

1  perfume_notes_classification = {
2      # CITRUS SMELLS
3      # Top Notes
4      "bergamot": "top", "bigarade": "top", "bitter orange": "top", "blood orange": "top",
5      "buddha's hand": "top", "calamansi": "top", "citron": "top", "citrus japonica": "top",
6      "citrus water": "top", "citruses": "top", ...

```

Şekil 3.1.1.1.2. Citrus Smell kategorisinde üst nota sınıflandırması.

3.1.1.2 Harici dosyadan nota yükleme mekanizması

Parfüm nota sözlüğünün harici bir dosyadan yüklenmesi, sistemin esnekliğini ve güncellenebilirliğini artıran önemli bir özelliktir. Bu mekanizma, veri setinin genişletilmesine ve parfüm uzmanları tarafından belirli kategorilere yeni notaların

eklenmesine olanak tanır. Çalışmamızda kullanılan harici dosyadan nota yükleme mekanizması aşağıdaki özelliklere sahiptir.

Sistem, CSV (Comma-Separated Values) formatında hazırlanmış nota listeleri ile çalışmaktadır. Bu format seçimi, parfüm notalarının tablolaştırma programlarında (Excel, Google Sheets vb.) düzenlenmesine ve kolayca CSV formatına aktarılabilmesine olanak tanır. Dosya yapısı şu şekildedir:

- Her satır bağımsız nota isimlerini içerir
- Virgülle ayrılmış değerler formatında düzenlenmiştir
- UTF-8 kodlaması kullanılarak dil karakterlerinin doğru işlenmesi sağlanmıştır

Sistem, dosya okuma sırasında oluşabilecek çeşitli hatalara karşı dayanıklı olacak şekilde tasarlanmıştır:

1. Dosya bulunamazsa varsayılan nota listesi kullanılır
2. Dosya açılırken veya okunurken hata oluşursa, hata mesajı gösterilir ve varsayılan listeye geri dönülür
3. Eksik veya boş satırlar otomatik olarak atlanır
4. Dosya formatı uygun değilse, sistem varsayılan notaları kullanmaya devam eder

```

1  if vocab_file and os.path.exists(vocab_file):
2      try:
3          with open(vocab_file, 'r', encoding='utf-8') as f:
4              import csv
5              reader = csv.reader(f)
6              for row in reader:
7
8                  self.notes.extend([item.strip().lower() for item in row if item.strip()])
9
10             self.notes = list(set(self.notes))
11             print(f"Sözlük başarıyla yüklendi. Toplam {len(self.notes)} nota.")
12         except Exception as e:
13             print(f"Sözlük yüklenirken hata: {e}")
14
15         self.notes = default_notes
16     else:
17
18         self.notes = default_notes
19
20
21     for i, note in enumerate(self.notes):
22         self.note_to_idx[note] = i
23         self.idx_to_note[i] = note
24
25     def __len__(self):
26         return len(self.notes)
27
28     def get_index(self, note):
29         return self.note_to_idx.get(note.lower(), -1)
30
31     def get_note(self, idx):
32         return self.idx_to_note.get(idx, "unknown")
33
34     def get_all_notes(self):
35         return self.notes

```

Şekil 3.1.1.2.1. Harici dosyadan nota yükleme mekanizması

3.1.1.3. Hata yakalama ve varsayılan nota listesi kullanımı

“NoteDictionary” sınıfının “_init_” metodunda hata yakalama ve varsayılan nota listesinin kullanımı şu şekilde gerçekleştirilmiştir:

```

1  if vocab_file and os.path.exists(vocab_file):
2      try:
3          with open(vocab_file, 'r', encoding='utf-8') as f:
4              import csv
5              reader = csv.reader(f)
6              for row in reader:
7
8                  self.notes.extend([item.strip().lower() for item in row if item.strip()])
9
10             self.notes = list(set(self.notes))
11             print(f"Sözlük başarıyla yüklendi. Toplam {len(self.notes)} nota.")
12         except Exception as e:
13             print(f"Sözlük yüklenirken hata: {e}")
14
15         self.notes = default_notes
16     else:
17
18         self.notes = default_notes

```

Şekil 3.1.1.3.1. Hata yakalama ve varsayılan nota listesi kullanımı

Bu kodda:

1. İlk olarak “vocab_file” parametresinin geçerli olup olmadığı ve dosyanın var olup olmadığı kontrol edilir: “if vocab_file and os.path.exists(vocab_file)”
2. Dosya varsa, bir try-except bloğu içinde dosya açılır ve okunmaya çalışılır.
3. İki durumda varsayılan nota listesi (default_notes) kullanılır:
 - o Dosya açılırken veya okunurken bir hata oluşursa (except Exception as e)
 - o Dosya yolu geçersizse veya dosya mevcut değilse (else)
4. Her iki hata durumunda da kullanıcıya bilgi verilir ve program, varsayılan nota listesini kullanarak çalışmaya devam eder. Bu yaklaşım, programın sağlam (robust) olmasını sağlar.

3.1.2. Metin İşleme ve Nota Tespiti

Metin verileri, günümüzün bilgi çağında en yaygın ve değerli kaynaklardan biridir. Ancak, ham metin verileri doğrudan makine öğrenimi algoritmaları tarafından anlaşılabilir veya işlenemez. Bu nedenle, metinlerin sayısal vektörlere dönüştürülmesi,

yani vektörizasyon, Doğal Dil İşleme (NLP) alanında temel bir adımdır. Bu süreç, metinlerin anlamlı, işlenebilir matematiksel temsillerini oluşturarak, belge sınıflandırma, bilgi erişimi, öneri sistemleri ve anlamsal arama gibi çeşitli NLP görevlerinin kapısını aralar.1 Metin verilerini sayısal formatlara dönüştürme yeteneği, çeviri ve duygu analizi gibi birçok NLP görevi için kritik bir ön koşuldur. (Focal, 2024)

Metin verilerinin sayısal vektörlere dönüştürülmesi, yalnızca teknik bir ön işlem adımı olmanın ötesinde, dilin hesaplamalı olarak analiz edilmesini sağlayan temel bir paradigma değişimidir. Bu dönüşüm olmadan, gelişmiş algoritmaların dili anlaması ve işlemesi mümkün değildir. Dilin nicel olarak ifade edilebilmesi, tüm hesaplamalı dilbilim alanını mümkün kılar. Bu, basit bir veri dönüştürmenin ötesinde, makinelerin dilsel yapıları ve anlamı "görmesini" sağlayan bir köprü görevi görür. (Focal, 2024)

Bu rapor, metin vektörizasyonunda yaygın olarak kullanılan TF-IDF (Terim Sıklığı-Ters Belge Sıklığı) yöntemini, n-gram yaklaşımını (özellikle 1-2 gram), vektörler arası benzerliği ölçmede kullanılan Kosinüs Benzerliği ölçütünü ve bu benzerliklerin yorumlanmasında kritik rol oynayan benzerlik eşik değeri (threshold) seçiminin önemini detaylı bir şekilde incelemeyi amaçlamaktadır. Her bir tekniğin temel prensipleri, avantajları, sınırlamaları ve birbirleriyle olan etkileşimleri derinlemesine analiz edilecektir.

3.1.2.1. TF-IDF vektörizasyonu kullanımının nedenleri ve avantajları

TF-IDF (Term Frequency-Inverse Document Frequency), metin belgelerini sayısal vektörlere dönüştürmek için kullanılan istatistiksel bir yöntemdir. Bir terimin bir belgedeki sıklığı ile o terimin bir belge koleksiyonundaki nadirliğini dengeleyerek, kelimelerin bir belge için ne kadar önemli olduğunu nicel olarak belirler. Bu, özellikle bilgi erişimi, belge sınıflandırma ve öneri sistemleri gibi uygulamalarda metin analizine derinlemesine bir bakış açısı sunmakta . (Ji, 2025)

TF-IDF'nin temelini iki ana bileşen oluşturur:

Terim Sıklığı (TF): Bir terimin belirli bir belgede ne sıklıkta görüldüğünü ölçer. Bir terim belgede ne kadar çok görünürse, terim sıklığı o kadar yüksek olur. Bu, kelimenin o belge içindeki yerel önemini gösterir.

Ters Belge Sıklığı (IDF): Bir kelimenin daha geniş bir belge koleksiyonundaki önemini değerlendirir. Nadir kelimeler, yani daha az belgede görünen kelimeler, daha

yüksek bir IDF puanı olarak benzersizliklerini ve koleksiyon genelindeki ayırt edici önemlerini gösterir. (Rahul, 2024)

TF-IDF, metin verilerini anlamlı sayısal temsillere dönüştürmede çeşitli faydalar sunmaktadır:

- Anahtar Terimlerin Belirlenmesi
- Belge Sınıflandırma ve Bilgi Erişimi Uygulamaları
- Dil Bağımsızlığı ve Hesaplama Verimliliği
- Boyut Azaltma ve Anlamsal İlgililik

TF-IDF'nin birçok avantajı olmasına rağmen, bazı sınırlamaları da mevcuttur:

- Anlamsal Anlayış Eksikliği
- Kısa Belgelerdeki Zorluklar ve Seyrek Veri Duyarlılığı

Hesaplama açısından verimli ve dilden bağımsız olmasına rağmen, TF-IDF'nin anlamsal anlayışındaki temel sınırlama, daha derin bağlamsal kavrayış veya nüans gerektiren görevler için daha gelişmiş tekniklerle (Word2Vec, BERT gibi) birleştirilmesi gerektiğini göstermektedir. Proje kapsamında "RoBERTa" modeli kullanılmış olup BERT modellemelerinden yararlanılmıştır.

3.1.2.2. N gram yaklaşımı (1-2 gram) kullanımı ile bileşik terimlerin yakalanması

Metin analizi, kelimeleri tekil birimler olarak ele alan "Bag of Words" (BoW) modelinin ötesine geçerek, kelimelerin dizilişini ve bağlamını yakalamak için n-gramları kullanır. N-gramlar, metindeki 'n' adet ardışık ögenin (kelime veya karakter) bir dizisidir ve doğal dilin zenginliğini ve karmaşıklığını daha iyi temsil etmeyi sağlar.

N-gramların Tanımı ve Türleri (Unigram, Bigram)

N-gramlar, 'n' değerine göre farklı türlerde adlandırılır:

- **Unigramlar:** n değerinin 1 olduğu n-gramlardır, yani tekil kelimeleri ifade eder. Metnin en temel birimleridir.
- **Bigramlar:** n değerinin 2 olduğu n-gramlardır, yani iki ardışık kelimenin dizisini ifade eder. Bu, kelime sırasını ve lokal bağlamı yakalamanın ilk adımıdır.⁹
- **Trigramlar ve Daha Fazlası:** n değerinin 3 olduğu trigramlar ve daha büyük n-gramlar, daha geniş bağlamları yakalar.

Shingles: Kelime n-gramları bazen "shingle" olarak da adlandırılır. (Alchemyleads, tarih yok)

Aşağıdaki tablo (Tablo 3.1.3.1.), n-gram türlerini ve örneklerini göstermektedir:

N-gram Türü	N Değeri	Tanım	Örnek Metin: "Follow good SEO practices"
Unigram	1	Tek kelime	("Follow", "good", "SEO", "practices") ^o
Bigram	2	İki ardışık kelime dizisi	("Follow good", "good SEO", "SEO practices") ^o
Trigram	3	Üç ardışık kelime dizisi	("Follow good SEO", "good SEO practices") ^o

Tablo 3.1.2.2.1. N gram türleri tablosu

“NoteFinder” sınıfında TF-IDF vektörizasyonu sırasında n-gram yaklaşımı kullanılmaktadır. İlgili kod parçası:

```

1 def __init__(self, note_dictionary):
2     self.note_dict = note_dictionary
3     self.vectorizer = TfidfVectorizer(analyzer='word', ngram_range=(1, 2), min_df=1)
4
5     self.note_vectors = self.vectorizer.fit_transform(self.note_dict.get_all_notes())

```

Şekil 3.1.2.2.1. Uygulama içerisinde N gram algoritması

“TfidfVectorizer” sınıfı “sklearn.feature_extraction.text” modülünden import edilmiştir. “ngram_range=(1, 2)” parametresi ile hem tek kelimeler (1-gram) hem de iki kelimelik bileşimler (2-gram) analiz edilmektedir.

Bu yaklaşım:

"vanilla" gibi tek kelimelik notaları (1-gram)

"black pepper", "blood orange" gibi iki kelimelik notaları (2-gram) analiz etmek için kullanılır.

Bu sayede:

"blood orange" ifadesi "blood" ve "orange" olarak ayrı ayrı değil, bütün bir terim olarak değerlendirilebilir.

Böylece koku notaları arasındaki anlam kaybı önlenir, çünkü "blood orange" tek başına "blood" veya "orange" ile aynı anlama gelmez.

“min_df=1” parametresi ise bir terimin en az bir dokümanda geçmesi gerektiğini belirtir, böylece tüm notalar vektörizasyon sürecine dahil edilmiştir.

3.1.2.3. Kosinüs benzerliği ölçütünün nota tespitindeki rolü

Metinler TF-IDF veya n-gramlar gibi yöntemlerle sayısal vektörlere dönüştürüldükten sonra, bu vektörler arasındaki benzerliği ölçmek için matematiksel bir ölçüte ihtiyaç duyulmaktadır bu nedenle Kosinüs Benzerliği ölçütü kullanılmıştır.

$$Similarity(A, B) = \frac{A \times B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad \text{Formül 3.1.2.3.1. Kosinüs Benzerliği}$$

“NoteFinder” sınıfının “find_notes” metodunda kosinüs benzerliği kullanılmaktadır. İlgili kod parçası ():

```

1 def find_notes(self, text, threshold=0.3, max_notes=10):
2     """Metinde semantik olarak benzer notaları tespit eder"""
3
4     if not isinstance(text, str) or not text.strip():
5         return []
6
7     text_vector = self.vectorizer.transform([text.lower()])
8
9
10    similarities = cosine_similarity(text_vector, self.note_vectors).flatten()
11
12
13    relevant_notes = []
14    for i, score in enumerate(similarities):
15        if score > threshold:
16            relevant_notes.append((self.note_dict.get_note(i), score))
17
18
19    relevant_notes.sort(key=lambda x: x[1], reverse=True)
20    return relevant_notes[:max_notes]

```

Şekil 3.1.2.3.1. Uygulama içerisinde Kosinüs Benzerliği algoritması

“sklearn.metrics.pairwise” modülünden import edilen “cosine_similarity” fonksiyonu kullanılıyor.

Kodda, metin vektörü (text_vector) ile tüm nota vektörleri (self.note_vectors) arasındaki kosinüs benzerliği hesaplanır.

1. “flatten()” metodu ile benzerlik skorları düzleştirilir ve bir dizi haline getirilir.
2. Bir eşik değeri (threshold) üzerindeki benzerlik skorlarına sahip notalar seçilir.

Bu yaklaşım ile bir metinde hangi parfüm notalarının bulunduğunu etkili bir şekilde tespit etmek için metin benzerliğini kullanılıp ve metindeki semantik içerikle parfüm notaları arasında ilişki kurması planlanmıştır.

3.1.2.3. Benzerlik eşik değeri (threshold) seçiminin önemi

Metin benzerliği puanları elde edildikten sonra, bir notanın "ilgili" veya "benzer" olarak kabul edilip edilmeyeceğine karar vermek için bir eşik değerine ihtiyaç duyulur. Bu eşik değeri, bir sınıflandırma modelinin çıktısını ikili bir karara (örneğin, "benzer" veya "benzer değil") dönüştüren kritik bir karar noktasıdır. Eşik değeri seçimi, sistemin performansını, özellikle de Hassasiyet (Precision) ve Geri Çağırma (Recall) metrikleri arasındaki dengeyi doğrudan etkiler. (Evidentlyai, 2025) Bu etki nedeniyle çoklu modelleme sistemine “Benzerlik Eşik Değeri” seçimi eklenmiştir. (Şekil 3.1.2.3.1.)

```

1 def find_notes(self, text, threshold=0.3, max_notes=10):
2     """Metinde semantik olarak benzer notaları tespit eder"""
3
4     if not isinstance(text, str) or not text.strip():
5         return []
6
7     text_vector = self.vectorizer.transform([text.lower()])
8
9
10    similarities = cosine_similarity(text_vector, self.note_vectors).flatten()
11
12    # Eşik değeri üzerindeki benzerlik skorlarına sahip notaları seçer
13
14    relevant_notes = []
15    for i, score in enumerate(similarities):
16        if score > threshold:
17            relevant_notes.append((self.note_dict.get_note(i), score))
18
19    # Benzerlik skoruna göre sıralar
20
21    relevant_notes.sort(key=lambda x: x[1], reverse=True)
22    return relevant_notes[:max_notes]

```

Şekil 3.1.2.3.1. Uygulama içerisinde Benzerlik Eşik Değeri algoritması

3.1.3. Tokenizasyon ve Metin Önışleme

3.1.3.1 RoBERTa tokenizer seçiminin gerekçesi

RoBERTa (Robustly Optimized BERT Pretraining Approach) tokenizer seçiminin gerekçeleri şunlardır:

1. **Gelişmiş Performans:** RoBERTa, daha fazla veri ve süreyle eğitildiği için BERT'e göre daha güçlü dil anlama yeteneklerine sahip olduğu araştırmalar sonucunda gözlemlenmiştir.
2. **BPE Tabanlı Tokenization:** Byte-Pair Encoding kullanarak nadir terimleri alt birimlere ayırır ve özel kelimeleri daha iyi işletebilir olmasını sağlaması proje çerçevesinde verimlilik sağlayacağı düşünüldüğü için seçilmiştir.
3. **Kontekst Duyarlılığı:** Kelimeleri bağlam içinde etkili şekilde yorumlayarak anlam derinliği sağlaması kelime gruplama stratejisinde destek olacağı öngörülmüştür.
4. **Çok Dilli Destek:** Farklı dil versiyonları sayesinde küresel metinleri analiz etmeye uygun olması projenin ilerleyen dönemlerinde geliştirme ve uluslararası çalışma sağlaması artı puan eklemiştir.
5. **Transfer Öğrenme Avantajı:** Özel alanlara uyarlanması kolaydır, az veriyle yüksek performans sunması işlem hızlarını artıracığı düşünüldüğü için seçilmiştir.
6. **Dynamic Masking:** Eğitim sırasında değişken maskeleyme ile metin ilişkilerini daha doğru kavrayacağı planlanmıştır.

```

1 self.text_encoder = RobertaModel.from_pretrained('roberta-base')
2 self.text_pooler = nn.Linear(768, 256)

```

Şekil 3.1.3.1. Uygulama içerisinde RoBERTa modeli kullanımını gösteren ilgili bölüm.

3.1.3.2 Tokenizasyon ve Metin Önişleme Sistem Stratejisi

RoBERTa ve diğer Transformer tabanlı modeller, giriş olarak maksimum 512 token kabul eder; bu nedenle “max_length=512” parametresi, tokenize edilen metni bu sınırdan tutar ve “truncation=True” ayarıyla uzun metinler otomatik olarak kesilir. Varsayılan kesme stratejisi metnin başından itibaren 512 token alıp geri kalanını atar; bu durum bilgi kaybına yol açabilir, ancak mimari gereklilik nedeniyle zorunludur. Neyse ki, parfüm incelemeleri genellikle kısa olduğu için bu sınırlama çoğu durumda sorun oluşturmaz. Ayrıca, “padding='max_length'” parametresiyle tüm metinler aynı uzunluğa getirilir; bu, “batch” işlemede “tensör” boyutlarının tutarlılığını sağlar. Kısa metinler, belirtilen uzunluğa ulaşana kadar özel padding token'ları ile doldurulur. Alternatif olarak, “padding='longest'” kullanılarak yalnızca en uzun metne göre doldurma yapılabilir; ancak max_length ile padding yapmak daha fazla bellek kullanmakla birlikte batch'ler arası daha yüksek tutarlılık sağlar. Bunun yanında, modelin boş veya geçersiz girişlerle karşılaşmaması için *if not text or not isinstance(text, str) or text.strip() == ""* gibi kontroller yapılır. Bu kontrolle None değeri, string tipi dışındaki girdiler ve yalnızca boşluk içeren ifadeler tespit edilir ve geçersiz durumlarda "Bu parfüm hakkında inceleme bulunmamaktadır." gibi bağlama uygun varsayılan bir metin atanır. Bu stratejiler, doğal dil işleme modellerinin güvenilir ve istikrarlı çalışmasını sağlamak için temel yaklaşımlar arasında yer alır. (Şekil 3.1.3.2.1.)

```
1 def prepare_batch(self, texts, max_length=512):
2     # Toplu metinleri tokenize et
3     batch_tokens = self.tokenizer(
4         texts,
5         max_length=max_length,
6         truncation=True,
7         padding='max_length',
8         return_tensors="pt"
9     )
10
11     return batch_tokens
```

Şekil 3.1.3.2.1. Uygulama içerisinde Tokenizasyon ve Metin Önleme Sistem Stratejisi.

3.2. Model Mimarisi ve Çok Kanallı Yapı

“PerfumeReviewMultiChannelModel” sınıfında, parfüm incelemelerini analiz etmek için çok kanallı bir mimari tasarlanmıştır. Bu mimarinin gerekçesi ve her kanalın özgün katkısı sistemin doğru çalışabilirliğini artırmak ve kullanıcı tarafında uygun çıktılar üretmek amacıyla seçilerek düzenlenmiş olup tasarlanmıştır. Model eksikliği ve hata durumları tespit edilerek projenin ileriki zamanlarda geliştirme çalışmalarında model ve yöntemlerde mimaride düzenlemeler meydana gelebilir. Ancak test aşamasında olan mimari metinsel, sayısal, kategorisel olarak kanallara ayrılarak modellenmiştir. (Şekil 3.2.1)

```
1 class PerfumeReviewMultiChannelModel(nn.Module):
2     def __init__(self, num_labels=5, num_notes=None, num_categories=5):
3         super().__init__()
4
5         if num_notes is None:
6             num_notes = 50
7
8         # Metin kanalı
9         self.text_encoder = RobertaModel.from_pretrained('roberta-base')
10        self.text_pooler = nn.Linear(768, 256)
11
12        # Sayısal kanal (parfüm özellikleri için)
13        self.numeric_encoder = nn.Sequential(
14            nn.Linear(10, 64),
15            nn.ReLU(),
16            nn.Dropout(0.2),
17            nn.Linear(64, 128),
18            nn.ReLU(),
19            nn.Dropout(0.2)
20        )
21
22        # Kategori kanalı
23        self.note_embeddings = nn.Embedding(num_notes, 64)
24        self.category_embeddings = nn.Embedding(num_categories, 64)
25        self.category_encoder = nn.Sequential(
26            nn.Linear(128, 128),
27            nn.ReLU(),
28            nn.Dropout(0.2)
29        )
```

Şekil 3.2.1. Model mimarisindeki kanallar.

Model içerisindeki kanallar Şekil 3.2.2 de gösterilen kod yapısı ile birleştirilmiştir.

```

1 def forward(self, text_ids, text_mask, numeric_features, notes, categories):
2     # Metin kodlama
3     text_output = self.text_encoder(text_ids, attention_mask=text_mask)
4     text_features = self.text_pooler(text_output.pooler_output)
5
6     # Sayısal kodlama
7     numeric_features = self.numeric_encoder(numeric_features)
8
9     # Kategori kodlama
10    note_features = self.note_embeddings(notes).mean(dim=1)
11    category_features = self.category_embeddings(categories).mean(dim=1)
12    category_combined = torch.cat([note_features, category_features], dim=1)
13    category_features = self.category_encoder(category_combined)
14
15    # Özellikleri birleştirme
16    combined = torch.cat([text_features, numeric_features, category_features], dim=1)
17
18
19    combined = combined.unsqueeze(0)
20    attn_output, _ = self.merger(combined, combined, combined)
21    attn_output = attn_output.squeeze(0)

```

Şekil 3.2.2. Model mimarisinde bulunan kanalların birleştirilmesi.

1. Metin Kodlama Kanalı:

- **Semantik Anlama:** RoBERTa modeli, parfüm incelemesindeki karmaşık dil yapılarını, metaforları ve duygusal ifadeleri anlar.
- **Kontekst Analizi:** Kelimelerin bağlamsal ilişkilerini yakalayıp parfüm notaları arasındaki ilişkileri tespit eder.
- **Zengin Temsil:** 768 boyutlu gömme vektörleri, metindeki nüansları yakalayıp daha kapsamlı bir analiz sağlar.
- **Transfer Öğrenme:** Önceden eğitilmiş dil modeli kullanarak, daha az veriyle bile yüksek performans elde edilir.

2. Sayısal Kodlama Kanalı:

- **Ölçülebilir Veriler:** Kalıcılık, yayılım, fiyat gibi sayısal değerleri işler.
- **Nesnel Ölçütler:** Metin analizinin öznel doğasını, nesnel sayısal değerlerle dengeler.
- **Doğrusal Olmayan Dönüşümler:** ReLU aktivasyonları sayesinde karmaşık sayısal ilişkileri modelleyebilir.
- **Düzenleştirme:** Drop katmanları ile aşırı öğrenmeyi önleyerek modelin genelleme yeteneğini artırır.

3. Kategori Kanalı:

- **Yapılandırılmış Bilgi:** Parfüm notaları ve kategorileri gibi taksonomik bilgileri modele entegre eder.
- **İlişkisel Öğrenme:** Notalar arasındaki ilişkileri öğrenerek daha tutarlı tahminler yapar.
- **Domain Expertise:** Parfümeri alanına özgü bilgileri ve hiyerarşileri doğrudan modele dahil eder.
- **Sparse Veri İşleme:** Kategorik verileri dense embedding'lere dönüştürerek etkin şekilde işler.

4. Birleştirme:

- **Çapraz Kanal Etkileşimi:** Multi-head attention mekanizması, farklı kanallardan gelen bilgilerin birbiriyle etkileşimini sağlar
- **Kanallar Arası Ağırlıklandırma:** Her analiz için en alakalı kanalların otomatik olarak vurgulanmasını sağlar
- **Özellik Artırma:** Kanalların birbirlerini tamamlayarak daha kapsamlı bir temsil oluşturmaya olanak tanır
- **Entegre Öğrenme:** Farklı veri türlerinin entegrasyonu ile daha doğru ve anlamlı tahminler yapılabilir

Bu çok kanallı mimari, parfüm incelemelerinin doğal dili, sayısal özellikleri ve kategorik bilgileri bir araya getirerek daha kapsamlı ve hassas bir analiz alınmasını sağlar.

3.2.1. RoBERTa Tabanlı Metin Kodlama: Pre-trained avantajları, havuzlama stratejisinin etkileri

RoBERTa modelinin seçim gerekçeleri "RoBERTa tokenizer seçiminin gerekçesi" adlı başlık altında açıklanmıştır burada ise açıklanacak olan ön eğitim avantajları ve havuzlama stratejisi hakkında olacaktır.

Önceden eğitilmiş modelleri kullanmanın, başkalarının bilgi ve deneyiminden yararlanma yeteneği, zamandan ve kaynaklardan tasarruf etme yeteneği ve model performansını iyileştirme yeteneği gibi çeşitli avantajları vardır. Önceden eğitilmiş modeller genellikle büyük, çeşitli veri kümeleri üzerinde eğitilir ve çok çeşitli desenleri ve özellikleri tanımak üzere eğitilmiştir. Sonuç olarak, ince ayar için güçlü bir temel sağlayabilir ve modelin performansını önemli ölçüde iyileştirebilir. (Sözlüğü, Önceden Eğitilmiş Model, tarih yok).

Transfer Öğrenme: RoBERTa, milyarlarca kelime üzerinde önceden eğitilmiştir. Bu sayede:

- Parfümeri alanına özgü sınırlı veri seti ile bile etkili sonuçlar elde edilebilir
- "Ahşap", "meyvemsi", "çiçeksi" gibi sıfatların semantik anlamları zaten modelce bilinmektedir

Bağlamsal Anlama: Pre-trained model, kelimelerin bağlama göre anlamlarını kavrar:

- "Güçlü bir açılış" ifadesinin parfüm bağlamında ne anlama geldiğini anlayabilir
- "Sert", "ağır", "hafif" gibi subjektif terimlerin parfüm bağlamındaki anlamlarını çözebilir

Dil Yapısı Bilgisi: Model, doğal dil yapısını önceden öğrenmiştir:

- Cümle yapısı, dilbilgisi ve sözdizimi bilgisi ile parfüm incelemelerini daha iyi analiz eder
- İnceleme metinlerinde gizli kalmış bilgileri ve ima edilen notaları tespit edebilir

Hesaplama Verimliliği: Sıfırdan eğitime kıyasla:

- Çok daha az eğitim süresi gerektirir
- Daha az eğitim verisi ile yüksek performans sağlar

Sinir ağlarında, bir havuzlama katmanı, birçok vektör arasında dağılmış bilgileri daha az sayıda vektöre indirgeyen ve toplayan bir tür ağ katmanıdır. Birkaç kullanımı vardır . Gereksiz bilgileri kaldırır, gereken hesaplama ve bellek miktarını azaltır, modeli girdideki küçük değişikliklere karşı daha sağlam hale getirir ve ağdaki sonraki katmanlardaki nöronların alıcı alanını artırır. (contributors, 2025)



Şekil 3.2.1.1 Model mimarisinde havuz stratejisi

3.2.1.1 Havuzlama (Pooling) Stratejisinin Etkileri:

1. **Boyut Azaltma:** RoBERTa'nın 768 boyutlu çıktısı, 256 boyuta indirgeniyor:
 - [self.text_pooler = nn.Linear\(768, 256\)](#)
 - Bu, modelin hesaplama verimini artırır
 - Diğer kanalların boyutlarıyla uyum sağlar (sayısal kanal: 128, kategori kanalı: 128)
2. **Özellik Yoğunlaştırma:** Havuzlama işlemi, RoBERTa'nın ürettiği zengin temsilleri daha kompakt hale getirirken:
 - En önemli özellikleri korumayı amaçlar
 - Gürültüyü azaltır ve sinyali güçlendirir

3. **Temsil Dengeleme:** Farklı uzunluktaki incelemeleri sabit boyutlu bir vektöre dönüştürerek:
 - Uzun ve kısa incelemeler arasında adil bir karşılaştırma sağlar
 - Model, incelemenin uzunluğundan çok içeriğine odaklanabilir
4. **CLS Token Kullanımı:** RoBERTa'nın "pooler_output" özelliği, özel CLS token'ına dayalı bir temsildir:
 - Bu token, tüm metin dizisinin özetleyici bir temsilini içerir
 - Parfüm incelemesinin genel tonunu ve ana temalarını yakalar

Bu havuzlama stratejisi, RoBERTa'nın ürettiği zengin metin temsillerini, çok kanallı modelin diğer kanallarıyla etkili bir şekilde entegre edilebilecek bir formata dönüştürürken bilgi kaybını en aza indirmeyi amaçlanmaktadır.

3.2.2. Sayısal ve Kategorik Özellikler

Parfüm incelemelerinde, metin dışında sayısal ve kategorik özelliklerin işlenmesi için model özel mimari bileşenler kullanmaktadır. Bu özellikler aşırı öğrenmeyi (overfitting) önleyen çeşitli tekniklerle işlenir.

```

1 self.numeric_encoder = nn.Sequential(
2     nn.Linear(10, 64),
3     nn.ReLU(),
4     nn.Dropout(0.2),
5     nn.Linear(64, 128),
6     nn.ReLU(),
7     nn.Dropout(0.2)
8 )

```

Şekil 3.2.2.1 Sayısal verilerin kodlanma mimarisi

Sayısal Veri İşleme Özellikleri:

1. **Çok Katmanlı İşleme:** İki lineer katman (10→64→128) kullanılarak sayısal verilerdeki karmaşık ilişkiler yakalanır
 - Kalıcılık, yayılım, fiyat gibi parfüm özellikleri arasındaki doğrusal olmayan ilişkiler öğrenilebilir
2. **ReLU Aktivasyonu:** Doğrusal olmayan aktivasyon fonksiyonu ile:
 - Sayısal özelliklerin doğrusal olmayan dönüşümü sağlar
 - Gradyan sorunu olmadan derin ağların eğitimi mümkün olur
 - Pozitif değerler korunurken negatif değerler sıfırlanır (parfüm puanları gibi pozitif değerler için uygundur)
3. **Boyut Genişletme:** Başlangıçta 10 boyutlu sayısal özellikler, 128 boyuta genişletilir
 - Bu, sayısal özelliklerin daha zengin bir temsil kazanmasını sağlar
 - Özellikler arası ilişkilerin daha kapsamlı öğrenilmesine olanak tanır

Kategorik Verilerin Gömülü Katmanları:

```

1  # Kategori kodlama
2  note_features = self.note_embeddings(notes).mean(dim=1)
3  category_features = self.category_embeddings(categories).mean(dim=1)
4  category_combined = torch.cat([note_features, category_features], dim=1)
5  category_features = self.category_encoder(category_combined)
6
7  # Kategori kanalı
8  self.note_embeddings = nn.Embedding(num_notes, 64)
9  self.category_embeddings = nn.Embedding(num_categories, 64)
10 self.category_encoder = nn.Sequential(
11     nn.Linear(128, 128),
12     nn.ReLU(),
13     nn.Dropout(0.2)
14 )

```

Şekil 3.2.2.1 Kategori verilerinin gömülme mimarisi

Embedding Dönüşümü: One-hot encoding yerine öğrenilebilir embedding vektörleri kullanılır

Her nota ve kategori için 64 boyutlu yoğun vektörler oluşturulur

Notalar arası semantik benzerlikler öğrenilebilir (örn. turuncgil notaları birbirine yakın olabilir)

Ortalama Alınması: “mean(dim=1)” operasyonu ile deęişken sayıda notanın sabit boyutlu temsili oluşturulur bu sayede:

- Bir parfümde bulunan tüm notaların ortalama temsili elde edilir
- Bu, farklı sayıda notaya sahip parfümleri karşılaştırılabilir hale getirir

Birleştirme ve İşleme: Nota ve kategori temsilleri birleştirilip ek bir dönüşüm uygulanır

- “torch.cat([note_features, category_features], dim=1)” ile birleştirme
- Kategorik özelliklerin birbiriyle olan ilişkileri öğrenilebilir

Dropout ve Overfitting Önleme Yöntemleri

Modelde aşırı öğrenmeyi önlemek için birkaç stratejik nokta bulunmaktadır:

a. Dropout Katmanları

Sayısal kanalda: İki dropout katmanı ($p=0.2$) ile nöronların %20'si rastgele devre dışı bırakılır

Kategori kanalında: Kategorik özelliklerin işlenmesinde de dropout ($p=0.2$) uygulanır

Etkileri:

- Model, herhangi bir özelliğe aşırı bağımlı olmamayı öğrenir
- Eğitim sırasında farklı özellik kombinasyonları denenmiş olur
- Test aşamasında daha genelleştirilebilir sonuçlar elde edilir

b. Parametrelerin Paylaşımı

Embedding katmanları, aynı kategorik değerleri her yerde aynı şekilde temsil eder bu olay da modelin öğrenmesi gereken parametre sayısını azaltır

c. Mimari Tasarım Tercihleri

- Boyut Sınırlama: Embedding boyutlarının 64 ile sınırlanması
- Katman Sayısı: Az sayıda katman kullanılması

Özellik Birleştirme: Farklı kanallardaki özelliklerin birleştirilmesiyle modelin çapraz doğrulama yapabilmesi

Bu dropout ve overfitting önleme teknikleri, parfüm incelemelerinin çok fazla varyasyon içerebileceği ve veri setinin sınırlı olabileceği durumlarda, modelin gerçek dünyada henüz görmediği incelemeler üzerinde de başarılı olmasını sağlar.

3.2.3. Özellik Birleştirme ve Dikkat Mekanizması

Modelin çeşitli kanallardan gelen bilgileri etkili şekilde birleştirmesi için kullanılan dikkat mekanizması, modern derin öğrenme mimarilerinin en güçlü bileşenlerinden biri olması nedeniyle kullanılmıştır.

```

1 # Birleştirme ve Dikkat Mekanizması
2 self.merger = nn.MultiheadAttention(embed_dim=512, num_heads=8)
3 self.norm1 = nn.LayerNorm(512)
4 self.norm2 = nn.LayerNorm(512)
5
6 #-----
7
8 # Özellikleri birleştirme
9 combined = torch.cat([text_features, numeric_features, category_features], dim=1)
10
11 # Multi-head attention ile özellikleri işleme
12 combined = combined.unsqueeze(0) # [batch_size, seq_len=1, embed_dim]
13 attn_output, _ = self.merger(combined, combined, combined)
14 attn_output = attn_output.squeeze(0) # [batch_size, embed_dim]
15
16 # Residual bağlantı ve normalizasyon
17 attn_output = combined.squeeze(0) + attn_output # Residual bağlantı
18 attn_output = self.norm1(attn_output) # Katman normalizasyonu

```

Şekil 3.2.3.1 Özellik Birleştirme ve Dikkat Mekanizması

- **8 Başlık (Heads):** Dikkat mekanizması 8 farklı alt uzayda paralel olarak çalışır.
- **Her Başlık:** $512/8 = 64$ boyutlu bir alt uzayda dikkat hesaplar.
- **Alt Uzaylar:** Farklı başlıklar, farklı türdeki ilişkilere odaklanabilir:
 - Bir başlık parfüm notaları ile fiyat arasındaki ilişkiye.
 - Diğer bir başlık notalar ile kullanıcı yorumları arasındaki ilişkiye.
 - Başka bir başlık ise yorum içindeki duygusal ifadelerle parfüm özellikleri arasındaki ilişkiye odaklanabilir.
- **Paralel İşleme:** 8 başlık sayesinde, bilginin farklı yönleri eşzamanlı olarak işlenir.

3.2.3.1. Residual Bağlantılar:

“`attn_output = combined.squeeze(0) + attn_output # Residual bağlantı`” kodu ile gerçekleştirilen bağlantı şu şekilde açıklanabilir:

- **Doğrudan Bağlantı:** Orijinal birleştirilmiş özellikler (combined), dikkat çıktısına (attn_output) doğrudan eklenir
- **Bilgi Akışı:** Dikkat mekanizmasından geçmeyen ham bilginin korunmasını sağlar
- **Gradyan Akışı:** Derin ağlarda gradyan kaybolması sorununu azaltır
- **Özellik Zenginleştirme:** Hem ham özellikleri hem de dikkat ile işlenmiş özellikleri bir arada tutar
- **Etki:** Modelin daha stabil eğitilmesini ve daha derin mimarilere olanak sağlar

3.2.3.2. Katman Normalizasyonu:

```
self.norm1 = nn.LayerNorm(512)
attn_output = self.norm1(attn_output)
```

Kodları ile oluşturulan normalizasyon yapısı:

- **İstatistiksel Dengeleme:** Her örnek için özellik değerlerini normalleştirir (ortalama 0, standart sapma 1)
- **Eğitim Stabilitesi:** Derin ağlarda aktivasyon değerlerinin patlama/sönme sorununu engeller
- **Mini-batch Bağımsız:** Her örneği bağımsız olarak normalleştirir, bu da batch boyutundan etkilenmemesini sağlar
- **İç Kovaryans Kayması:** Farklı katmanlar arasındaki aktivasyon dağılımı değişimlerini azaltır
- **Hızlı Yakınsama:** Optimizasyon sürecinin hızlanmasını sağlar

Ve bu sayede katmanlar arası normalizasyon sağlanmış olur.

3.2.3.3 Birleştirilen Bilginin Katkısı

- **Çok Modlu Füzyon:** Üç farklı modalitedeki bilgiyi (metin, sayısal, kategorik) tek bir temsilde birleştirir
- **Özellik Zenginleştirme:** Her kanalın katkısı korunarak daha kapsamlı bir temsil oluşturulur
- **Çapraz-Kanal İlişkileri:** Dikkat mekanizması sayesinde farklı kanallar arasındaki karmaşık ilişkiler öğrenilebilir

- **Adaptif Ağırlıklandırma:** Model, her örnek için hangi kanalın daha önemli olduğunu öğrenebilir
- **Holistik Analiz:** Parfüm incelemelerinin farklı yönlerini bütünsel olarak değerlendirme imkânı sağlar.

Bu dikkat mekanizması ve birleştirme yaklaşımı, parfüm incelemelerinin karmaşık doğasını analiz etmek için farklı veri türlerinden elde edilen bilgileri etkili bir şekilde entegre eder. Her kanal kendi güçlü yönleri ile katkıda bulunurken, multi-head attention mekanizması bu bilgileri optimal şekilde harmanlayarak sisteme en iyi şekilde destek sağlar.

3.3 Model Girdileri ve Çıktıları

3.3.1. Metin ve Özellik Girişi

Sistemde kullanılan metinler, not veritabanından alınarak TF-IDF vektörizasyon işlemine tabi tutulmuştur. Kullanıcı tarafından girilen sorgular ise aynı vektör uzayına dönüştürülerek, korpustaki notlarla semantik benzerlik açısından karşılaştırılmaktadır. Bu süreçte kelime frekansı ve ters belge frekansı bilgileri kullanılarak metinlerin ayırt edici özellikleri belirlenmiştir.

3.3.1.1 RoBERTa Tokenizer Kullanımı

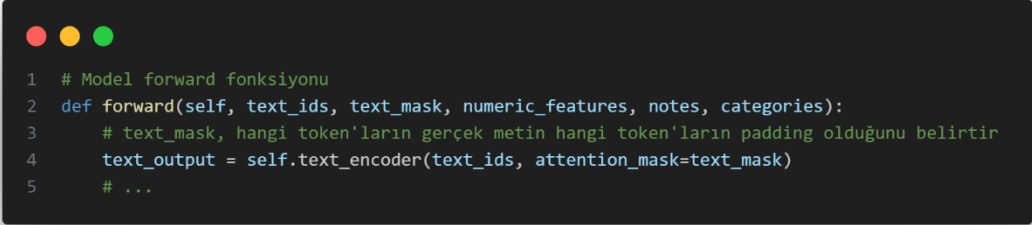
```
1 def tokenize_review(self, review_text):
2
3     tokens = self.tokenizer(
4         review_text,
5         max_length=512,
6         truncation=True,
7         padding='max_length',
8         return_tensors='pt'
9     )
10
11     return tokens.input_ids, tokens.attention_mask
```

Şekil 3.3.1.1.1 RoBERTa Tokenizer Kullanımı

İşleme sırasıyla aşağıdaki gibi uygulanmıştır.

1. **Tokenizasyon:** RoBERTa'nın BPE (Byte-Pair Encoding) tabanlı tokenizer'ı kullanılır
2. **Maksimum Uzunluk:** 512 token ile sınırlandırılır (RoBERTa'nın mimari limiti)
3. **Kesme (Truncation):** Uzun incelemeler 512 token ile sınırlandırılır
4. **Padding:** Kısa incelemeler, maksimum uzunluğa özel padding token'ları ile doldurulur
5. **Tensor Dönüşümü:** Tokenize edilmiş girdi PyTorch tensor'larına dönüştürülür

3.3.1.2 Girdi maskeleri



```

1 # Model forward fonksiyonu
2 def forward(self, text_ids, text_mask, numeric_features, notes, categories):
3     # text_mask, hangi token'ların gerçek metin hangi token'ların padding olduğunu belirtir
4     text_output = self.text_encoder(text_ids, attention_mask=text_mask)
5     # ...

```

Şekil 3.3.1.2.1 Girdi maskeleri için kullanılan fonksiyonlar

- **Attention Mask:** text_mask parametresi, gerçek token'ları (1) ve padding token'larını (0) ayırt eder
- **Maskenin Etkisi:** RoBERTa modeli, attention hesaplarırken padding token'larını görmezden gelir
- **Verimlilik:** Maskeleme sayesinde model hesaplama gücünü yalnızca gerçek içeriğe odaklar
- **Değişken Uzunluk:** Farklı uzunluktaki incelemeler için batch işlemeyi mümkün kılar

3.3.1.3.Ön İşleme Adımları

```

1  def preprocess_review(self, text):
2      # Metin temizleme
3      text = re.sub(r'^\w\s', ' ', text) # Noktalama işaretlerini kaldır
4      text = re.sub(r'\s+', ' ', text).strip() # Fazla boşlukları temizle
5
6      # Metin küçültme
7      text = text.lower()
8
9      # Özel karakter normalizasyonu
10     text = unicodedata.normalize('NFKD', text)
11
12     return text

```

Şekil 3.3.1.3.1. Ön İşleme Adımları için kullanılan fonksiyonlar

- **Temizleme:** Noktalama işaretleri ve özel karakterler temizlenir
- **Normalizasyon:** Metinler küçük harfe çevrilir, boşluklar standardize edilir
- **Unicode İşleme:** Aksanlı karakterler normalize edilir
- **Tutarlılık:** Bu adımlar, modelin farklı formatlardaki metinleri tutarlı şekilde işlemesini sağlar

3.3.1.4. Sayısal ve Kategorik Bilgiler

```

1  # Sayısal özellikler (10 boyutlu)
2  numeric_features = torch.tensor([
3      perfume.longevity, # Kalıcılık
4      perfume.sillage, # Yayılım
5      perfume.price_value, # Fiyat/Değer oranı
6      perfume.versatility, # Çok yönlülük
7      perfume.packaging, # Ambalaj
8      perfume.seasonality, # Mevsimsellik
9      perfume.gender_score, # Cinsiyet puanı (0: erkek, 0.5: unisex, 1: kadın)
10     perfume.overall_rating, # Genel değerlendirme
11     perfume.complexity, # Karmaşıklık
12     perfume.uniqueness # Benzersizlik
13 ], dtype=torch.float32)
14
15 # Kategorik özellikler - notalar ve kategoriler (ID'ler)
16 notes = torch.tensor([note_dict.get_index(note) for note in perfume.notes], dtype=torch.long)
17 categories = torch.tensor([category_dict.get_index(cat) for cat in perfume.categories], dtype=torch.long)

```

Şekil 3.3.1.4.1. Sayısal ve Kategorik Bilgiler için kullanılan fonksiyonlar

- **Sayısal Özellikler:** 10 boyutlu bir vektörde parfümün sayısal değerlendirmeleri tutulur
- **Nota ID'leri:** Parfümün içerdiği her nota, sözlük ID'sine çevrilerek bir tensor'da saklanır
- **Kategori ID'leri:** Parfümün ait olduğu her kategori (örn. Chypre, Fougere) ID'lere çevrilir
- **Veri Tipleri:** Sayısal veriler için “torch.float32”, kategorik ID'ler için “torch.long” kullanılır

3.3.1.5. Embedding'lerin Modele Beslenmesi

```

1 # Kategori kodlama
2 note_features = self.note_embeddings(notes).mean(dim=1) # [batch_size, 64]
3 category_features = self.category_embeddings(categories).mean(dim=1) # [batch_size, 64]

```

Şekil 3.3.1.5.1. Embedding'lerin Modele Beslenmesi için kullanılan kodlar.

- **Embedding Tablosu:** Her nota ve kategori ID'si, öğrenilebilir bir embedding vektörüne (64 boyutlu) dönüştürülür
- **Ortalama Alma:** Bir parfüme ait tüm notaların embedding'leri ortalaması alınır
- **Sabit Boyut:** Farklı sayıda nota içeren parfümleri sabit boyutlu bir vektörle temsil etmeyi sağlar
- **Öğrenilebilir İlişkiler:** Embedding'ler eğitim sırasında güncellenir, böylece model notalar arasındaki ilişkileri öğrenebilir

3.3.1.6. Batch Yönetimi

```

1 def collate_batch(self, batch):
2     # Batch içindeki örneklerin unpacking işlemi
3     texts, numeric_feats, note_ids, category_ids, labels = zip(*batch)
4
5     # Metinleri tokenize et ve batch'le
6     encoded_texts = self.tokenizer(
7         list(texts),
8         padding='max_length',
9         truncation=True,
10        max_length=512,
11        return_tensors='pt'
12    )
13
14    # Diğer özellikleri batch'le
15    numeric_batch = torch.stack(numeric_feats)
16
17    # Farklı uzunluktaki nota ID listelerini padding ile eşitle
18    max_notes = max(len(notes) for notes in note_ids)
19    padded_notes = torch.zeros((len(note_ids), max_notes), dtype=torch.long)
20    for i, notes in enumerate(note_ids):
21        padded_notes[i, :len(notes)] = torch.tensor(notes)
22
23    # Benzer şekilde kategorileri de padding ile eşitle
24    max_cats = max(len(cats) for cats in category_ids)
25    padded_cats = torch.zeros((len(category_ids), max_cats), dtype=torch.long)
26    for i, cats in enumerate(category_ids):
27        padded_cats[i, :len(cats)] = torch.tensor(cats)
28
29    # Etiketleri tensor'a dönüştür
30    labels_batch = torch.tensor(labels)
31
32    return encoded_texts.input_ids, encoded_texts.attention_mask, numeric_batch, padded_notes, padded_cats, labels_batch

```

Şekil 3.3.1.6.1. Batch Yönetimi için kullanılan kodlar

- **Batch Oluşturma:** Farklı örnekleri tek bir batch'e birleştirir
- **Dinamik Padding:** Batch içindeki en uzun nota/kategori listesine göre diğerleri doldurulur
- **Tensor Dönüşümü:** Tüm girdiler PyTorch tensor'larına dönüştürülür
- **Bellek Verimli:** Padding işlemi, batch işlemeyi mümkün kılarken bellek kullanımını optimize eder

3.3.1.7. Çıktılar

```

1 # Çıktı katmanları
2 def forward(self, text_ids, text_mask, numeric_features, notes, categories):
3     # ...işleme adımları...
4
5     # Sınıflandırma çıktısı (duygu analizi)
6     sentiment_logits = self.sentiment_classifier(attn_output)
7
8     # Regresyon çıktıları
9     regression_hidden = F.relu(self.regressor_hidden(attn_output))
10    regression_outputs = self.regressor(regression_hidden)
11
12    # Nota sentezi çıktısı
13    notes_output = self.note_synthesizer(attn_output)
14
15    return {
16        'sentiment': sentiment_logits,
17        'regression': regression_outputs,
18        'notes': notes_output
19    }

```

Şekil 3.3.1.7.1. Çıktı için kullanılan kodlar.

- **5 Sınıf:** Parfüm incelemesinin duygusal tonunu 5 kategoride sınıflandırır (çok olumsuz, olumsuz, nötr, olumlu, çok olumlu)
- **Logits:** Çıktı, henüz softmax uygulanmamış ham logit'lerdir
- **Cross-Entropy:** Eğitimde genellikle cross-entropy loss ile bu logit'ler işlenir

Regresyon (Parfüm Özellikleri)

- **5 Özellik:** Parfümleri sayısal olarak değerlendiren 5 özellik tahmin edilir
- **Gizli Katman:** İlk olarak 256 boyutlu bir gizli katman kullanılır
- **ReLU Aktivasyonu:** Doğrusal olmayan dönüşüm için ReLU aktivasyonu uygulanır
- **MSE Loss:** Eğitimde genellikle Mean Squared Error loss kullanılır

Nota Sentezi (Sigmoid Katman)

- **Multi-label Çıktı:** Her parfüm notasının varlığını/yokluğunu tahmin eder
- **Sigmoid Aktivasyonu:** Her notanın var olma olasılığını 0-1 arasında bir değere dönüştürür

- **Bağımsız Tahminler:** Her nota bağımsız olarak tahmin edilir (bir parfümde birden fazla nota olabilir)
- **Binary Cross-Entropy:** Eğitimde her nota için binary cross-entropy loss kullanılır
- **Threshold ile Seçim:** Tahminler arasından belirli bir eşik değerinin üzerindeki seçilir

Bu çok görevli mimari, parfüm incelemelerinden çeşitli bilgiler (duygu analizi, özelliklerin regresyonu ve içerik notalarının tahmini) çıkararak kapsamlı bir parfüm analizi sağlar.

3.4. Parfüm Analiz Sistemi

Geliştirilen parfüm analiz sistemi, farklı parfüm notalarının kimyasal ve duysal özelliklerini TF-IDF vektörizasyonu ile sayısallaştırarak benzerlik tespiti yapmaktadır. Kullanıcı tarafından girilen koku tanımlamaları, sistemdeki parfüm veritabanıyla karşılaştırılarak en uygun eşleşmeler belirlenmekte ve böylece tüketicilerin tercihlerine uygun parfüm önerileri sunulmaktadır.

3.4.1. PerfumeReviewAnalyzer Sınıfı

```

1 class PerfumeReviewAnalyzer:
2     def __init__(self, model_path=None, vocab_file=None):
3         # Not sözlüğünü oluşturma aşaması
4         try:
5             self.note_dict = NoteDictionary(vocab_file=vocab_file)
6             self.note_finder = NoteFinder(self.note_dict)
7         except Exception as e:
8             print(f"Not sözlüğü oluşturulurken hata: {e}")
9             # Varsayılan not sözlüğünü kullan
10            self.note_dict = NoteDictionary()
11            self.note_finder = NoteFinder(self.note_dict)
12
13            # Tokenizer oluşturma aşaması
14            try:
15                self.tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
16            except Exception as e:
17                print(f"Tokenizer yüklenirken hata: {e}")
18                raise RuntimeError("Tokenizer yüklenemedi. Lütfen transformers kütüphanesini kontrol edin.")
19
20            # Cihaz belirleme aşaması
21            self.device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
22            print(f"Kullanılan cihaz: {self.device}")
23
24            # Model oluşturma aşaması
25            self.model = PerfumeReviewMultiChannelModel(
26                num_labels=5,
27                num_notes=len(self.note_dict),
28                num_categories=5
29            ).to(self.device)
30
31            # Model yükleme aşaması
32            if model_path and os.path.exists(model_path):
33                try:
34                    checkpoint = torch.load(model_path, map_location=self.device)
35                    self.model.load_state_dict(checkpoint['model_state_dict'])
36                    print(f"Model başarıyla yüklendi: {model_path}")
37                except Exception as e:
38                    print(f"Model yüklenirken hata: {e}")
39                    print("Yeni model oluşturuldu.")
40            else:
41                print("Model dosyası bulunamadı veya belirtilmedi, yeni model oluşturuldu.")

```

Şekil 3.4.1. PerfumeReviewAnalyzer sınıfı yapısı

3.4.1.1. Model/Tokenizer Yükleme ve Hata Yönetimi

1. Katmanlı Hata Yakalama:

- Her bileşen (sözlük, tokenizer, model) ayrı try-except blokları ile yüklenir
- Herhangi bir hata durumunda alternatif çözümler devreye girer

2. Sözlük Yükleme Stratejisi:

- Belirtilen [vocab_file](#) kullanılmaya çalışılır
- Hata durumunda varsayılan nota sözlüğü kullanılır
- Bu yaklaşım, sistemin her durumda çalışmasını sağlar

3. Tokenizer Yükleme:

- RoBERTa tokenizer'ı doğrudan Hugging Face hub'dan yüklenir
- Yükleme başarısız olursa, işlem durdurulur (sistem için kritik bileşen)
- “RuntimeError” fırlatılarak uygulama seviyesinde kontrol sağlanır

4. Model Yükleme ve Kontrol:

- Önce model mimarisi oluşturulur
- Eğer model dosyası mevcutsa, ağırlıklar yüklenir
- Yükleme başarısız olursa, yeni başlatılan model kullanılır
- Bu sayede model eksikliği sistemin çalışmasını engellemez

3.4.2. Özellik İşleme ve Yorumlama

```

1  def preprocess_text(self, text):
2      # Metin kontrolü
3      if not isinstance(text, str) or not text.strip():
4          text = "Bu bir parfüm incelemesidir." # Varsayılan metin
5
6      # Tokenize et
7      encoded = self.tokenizer(
8          text,
9          max_length=512,
10         truncation=True,
11         padding='max_length',
12         return_tensors='pt'
13     )
14
15     return {
16         'input_ids': encoded['input_ids'].to(self.device),
17         'attention_mask': encoded['attention_mask'].to(self.device)
18     }

```

Şekil 3.4.2.1. Metin Ön İşleme yapısı

- **Giriş Doğrulama:** Boş veya geçersiz metin için varsayılan değer kullanımı
- **Tokenizasyon:** RoBERTa tokenizer'ı ile metin tokenize edilir
- **Sınırlama ve Doldurma:** 512 token limiti, kırpma ve maksimum uzunluğa göre doldurma
- **Tensor Dönüşümü:** PyTorch tensor'larına çevrilen veriler uygun cihaza taşınır

3.4.3. Nota Tespiti ve Özellik Hazırlama

```

1 def prepare_features(self, text):
2     # Metni tokenize etme işlemi
3     tokenized = self.preprocess_text(text)
4
5     # Notaları tespit etme işlemi
6     detected_notes, note_indices = self.note_finder.extract_notes_with_indices(text)
7
8     # Not indisleri tensor'a dönüştür ve padding yap
9     if note_indices:
10        notes_tensor = torch.tensor(note_indices, dtype=torch.long).to(self.device)
11    else:
12        notes_tensor = torch.zeros(1, dtype=torch.long).to(self.device)
13
14    # Numeric features
15    numeric_features = torch.zeros(1, 10, dtype=torch.float).to(self.device)
16
17    # Kategori bilgisi
18    categories = torch.zeros(1, dtype=torch.long).to(self.device)
19
20    return {
21        'text_ids': tokenized['input_ids'],
22        'text_mask': tokenized['attention_mask'],
23        'numeric_features': numeric_features,
24        'notes': notes_tensor.unsqueeze(0),
25        'categories': categories.unsqueeze(0),
26        'detected_notes': detected_notes
27    }

```

Şekil 3.4.3.1. Nota Tespiti ve Özellik Hazırlamaları için kod yapısı

- **Metinden Nota Tespiti:** “NoteFinder” ile metinden notalar çıkarılır
- **Boş Not Durumu Yönetimi:** Nota bulunamazsa sıfır tensörü oluşturulur
- **Placeholder Değerler:** Gerçek bir sistemde doldurulacak sayısal ve kategorik veriler için yer tutucular
- **Batch Boyutu Ekleme:** Tekil örnekleri batch formatına çevirir (unsqueeze(0))
- **Çok Kanallı Veri Yapısı:** Metin, sayısal ve kategorik veriler tek bir yapıda toplanır

3.4.4. Tahmin Sonuçlarının Değerlendirilmesi

```

1 def analyze(self, text):
2     # Model değerlendirme moduna geçilmesi
3     self.model.eval()
4
5     # Özelliklerin hazırlanması
6     features = self.prepare_features(text)
7
8
9     with torch.no_grad():
10        outputs = self.model(
11            features['text_ids'],
12            features['text_mask'],
13            features['numeric_features'],
14            features['notes'],
15            features['categories']
16        )
17
18        # Çıktıların işlenmesi
19        sentiment_logits = outputs['sentiment'].squeeze()
20        sentiment_probs = F.softmax(sentiment_logits, dim=0)
21        sentiment_class = torch.argmax(sentiment_probs).item()
22
23        regression_output = outputs['regression'].squeeze()
24
25        synthesized_notes = outputs['synthesized_notes'].squeeze()
26
27        # Kategori etiketleri
28        sentiment_categories = ['hate', 'dislike', 'neutral', 'like', 'love']
29        regression_categories = ['longevity', 'sillage', 'gender', 'price_value', 'quality']
30
31        # Sonuçların hazırlanması
32        results = {
33            'detected_notes': features['detected_notes'],
34            'sentiment': {
35                'label': sentiment_categories[sentiment_class],
36                'confidence': sentiment_probs[sentiment_class].item()
37            }
38        }
39
40        # Regresyon değerlerinin işlenmesi
41        for i, category in enumerate(regression_categories):
42            # 0-5 arasına normalize et
43            value = min(max(regression_output[i].item(), 0), 5)
44
45            # Cinsiyet tahmini için metin tabanlı tahmini de kullanılması
46            if category == 'gender':
47                # Model tahmini ile metin tabanlı tahmini birleştirilmesi
48                model_gender = value # Model tahmini
49                text_gender = determine_gender_from_text(text) # Metin tabanlı tahmin aşaması
50
51                # Ağırlıklı ortalamanın alınması
52                combined_gender = (model_gender * 0.6) + (text_gender * 0.4)
53                value = min(max(combined_gender, 0), 5)
54
55            results[category] = value
56
57        # Sentezlenen notaların işlenmesi
58        synthesized_note_scores = []
59        for i, score in enumerate(synthesized_notes):
60            if score.item() > 0.3:
61                synthesized_note_scores.append((self.note_dict.get_note(i), score.item()))
62
63        # Nota skorlarını sıralanması
64        synthesized_note_scores.sort(key=lambda x: x[1], reverse=True)
65        results['synthesized_notes'] = synthesized_note_scores[:10] # En iyi 10 nota
66
67        return results

```

Şekil 3.4.4.1. Tahmin Sonuçlarının Değerlendirilmesi için kod yapısı

- **Softmax Dönüşümü:** Ham logit'ler olasılık değerlerine dönüştürülür
- **Argmax ile Sınıf Seçimi:** En yüksek olasılıklı duygu kategorisi belirlenir
- **Etiket Ataması:** Sayısal indisler anlamlı metin etiketlerine (hate, dislike, neutral, like, love) dönüştürülür
- **Güven Skoru:** Seçilen kategorinin olasılık değeri güven skoru olarak kaydedilir

Regresyon Çıktılarının Ölçeklenmesi

- **Sınırlama:** Tüm değerler “ $\min(\max(\text{value}, 0), 5)$ ” ile 0-5 aralığına sınırlanır
- **Özel İşleme (Cinsiyet):** Cinsiyet tahmini için model çıktısı ve metin tabanlı analiz birleştirilir:
- **Hibrit Yaklaşım:** Model tahmini ve kural tabanlı tahminin ağırlıklı ortalaması alınır

Nota Skorlarının Filtrelenmesi ve Düzenlenmesi

- **Eşik Değeri Filtreleme:** Sadece 0.3'ten büyük skorlu notalar dikkate alınır
- **Sıralama:** Notalar güven skoruna göre azalan şekilde sıralanır
- **Sınırlama:** Sadece en iyi 10 nota sonuç olarak döndürülür

Bu çok kanallı sistem, parfüm incelemelerinin dilsel, sayısal ve kategorik özelliklerini bütünleşik bir şekilde işleyerek, parfüm özellikleri hakkında kapsamlı tahminler yapar ve bu tahminleri kullanıcı dostu bir formatta sunar.

3.5. Parfüm Tarifi ve Özellik Analizi

Kodda 300'den fazla parfüm notası üst (top), orta (middle) ve alt (base) nota kategorilerine sistematik olarak sınıflandırılmıştır. Sınıflandırılmamış yeni notalar için skor tabanlı bir tahmin mekanizması kullanılmaktadır.

```

1  if note_lower in perfume_notes_classification:
2      if perfume_notes_classification[note_lower] == "top":
3          top_notes.append((note, score))
4      elif perfume_notes_classification[note_lower] == "middle":
5          middle_notes.append((note, score))
6      elif perfume_notes_classification[note_lower] == "base":
7          base_notes.append((note, score))
8  else:
9      # Sınıflandırılmamış notalar için skor tabanlı karar ver
10     if score > 0.7: # Yüksek skorlu notalar alt nota olma eğiliminde
11         base_notes.append((note, score))
12     elif score > 0.5: # Orta skorlu notalar orta nota olma eğiliminde
13         middle_notes.append((note, score))
14     else: # Düşük skorlu notalar üst nota olma eğiliminde
15         top_notes.append((note, score))

```

Şekil 3.5.1. Skor tabanlı tahmin mekanizması

Bu yaklaşımın gerekçesi:

Üst notalar: Daha uçucu, daha az yoğun (düşük skorlu)

Orta notalar: Orta düzeyde kalıcılık (orta skorlu)

Alt notalar: Daha kalıcı, daha yoğun (yüksek skorlu)

3.5.1. Koku Özellikleri Dönüşümü

Sayısal regresyon çıktıları olan kalıcılık, yayılım ve cinsiyet değerleri (0-5 arası), anlamlı kategorik etiketlere dönüştürülmesi sağlanır.

```

1 # Değerleri kategorilere dönüştürülme aşaması
2 gender_categories = ['çok kadınsı', 'kadınsı', 'unisex', 'erkeksi', 'çok erkeksi']
3 longevity_categories = ['çok zayıf', 'zayıf', 'orta', 'uzun', 'çok uzun']
4 sillage_categories = ['çok hafif', 'hafif', 'orta', 'güçlü', 'çok güçlü']
5
6 # Değerleri kategorilere eşlenme aşaması
7 gender_idx = min(int(gender_value), 4)
8 longevity_idx = min(int(longevity_value), 4)
9 sillage_idx = min(int(sillage_value), 4)
10
11 gender_label = gender_categories[gender_idx]
12 longevity_label = longevity_categories[longevity_idx]
13 sillage_label = sillage_categories[sillage_idx]

```

Şekil 3.5.1.1. Koku özellikleri dönüşüm mekanizması

Bu dönüşüm sayesinde kullanıcı dostu bir açıklama sağlanır

- 0-1 arası: çok kadınsı/çok zayıf/çok hafif
- 1-2 arası: kadınsı/zayıf/hafif
- 2-3 arası: unisex/orta/orta
- 3-4 arası: erkeksi/uzun/güçlü
- 4-5 arası: çok erkeksi/çok uzun/çok güçlü

3.5.2. Tarif Oluşturma Mekanizması

```

1 # Her kategoriyi skora göre sırala
2 top_notes.sort(key=lambda x: x[1], reverse=True)
3 middle_notes.sort(key=lambda x: x[1], reverse=True)
4 base_notes.sort(key=lambda x: x[1], reverse=True)
5
6 # Her kategoriden en fazla 3-4 nota seç
7 max_notes_per_category = 4
8 top_notes = top_notes[:max_notes_per_category]
9 middle_notes = middle_notes[:max_notes_per_category]
10 base_notes = base_notes[:max_notes_per_category]

```

Şekil 3.5.2.1. Tarif oluşturma mekanizması

Profesyonel parfümler genellikle her kategoride sınırlı sayıda nota kullanır. Bu sınırlama:

- Dengelenmiş bir koku profili oluşturur
- Karmaşık ve çakışan notaları önler
- Endüstri standartlarına uygun bir formülasyon sağlar

3.5.3 Tarif Formatı ve Yapım Talimatları

```

1 # Tarifi oluştur
2 if top_notes:
3     recipe += "**Üst Notalar:** "
4     recipe += ", ".join([f"{note} ({score*100:.1f})" for note, score in top_notes])
5     recipe += "\n\n"
6
7 if middle_notes:
8     recipe += "**Orta Notalar:** "
9     recipe += ", ".join([f"{note} ({score*100:.1f})" for note, score in middle_notes])
10    recipe += "\n\n"
11
12 if base_notes:
13    recipe += "**Alt Notalar:** "
14    recipe += ", ".join([f"{note} ({score*100:.1f})" for note, score in base_notes])
15    recipe += "\n\n"
16
17 # Tavsiye edilen parfüm yapımı
18 recipe += "**Karıştırma Talimatları:**\n"
19 recipe += "1. Baz olarak p saf alkol ve 0 su karışımını hazırlayın.\n"
20
21 if top_notes:
22    recipe += "2. Üst notaları ekleyin ve karıştırın.\n"
23 if middle_notes:
24    recipe += "3. Orta notaları ekleyin ve karıştırın.\n"
25 if base_notes:
26    recipe += "4. Alt notaları ekleyin ve karıştırın.\n"
27
28 recipe += "5. Karışımı 2-4 hafta karanlık bir yerde dinlendirin, günde bir kez çalkalayın.\n"
29 recipe += "6. Filtreleyerek şişeleyin ve kullanıma hazırdır."

```

Şekil 3.5.2.1. Tarif Formatı ve Yapım Talimatları

Tarif formatındaki önemli özellikler:

1. **Notaların Önem Sırasıyla Listelenmesi:** Her kategori içinde, notalar güven skorlarına göre sıralanır
2. **Yüzde Bilgisi:** Her nota için güven skorunun yüzde olarak gösterimi, hangi notaların daha baskın olması gerektiğine dair rehberlik sağlar
3. **Kategori Bazlı Yapılandırma:** Üst, orta ve alt notalar ayrı bölümlerde listelenir
4. **Dinlendirme Süreci:** Profesyonel parfüm yapım süreçlerine uygun olarak, karışımın 2-4 hafta dinlendirilmesi önerilir
5. **Adım Adım Talimatlar:** Kullanıcıya karıştırma, dinlendirme ve filtreleme adımlarını içeren pratik yönergeler sunulur

Bu yaklaşım, kullanıcıya sadece notaları listelemekle kalmayıp, bu notalarla gerçekten bir parfüm yapabileceği pratik bir tarif sağlar. Ayrıca, inceleme metninden elde edilen cinsiyet, kalıcılık ve yayılım gibi özellikleri de tarif içinde konumlandırarak, incelenen parfümün karakterine benzer bir formül oluşturmak amaçlanmaktadır.

3.6. Cinsiyet Analizi ve Tahmin Birleşimi

Terim Tabanlı Yöntem

Parfüm incelemelerinde cinsiyet yönelimini belirlemek için terim eşleştirme ve ağırlıklandırılmış puanlama yaklaşımı kullanılmaktadır ve kullanılan kod Şekil 3.6.1 de gösterilmiştir.

```

1 def determine_gender_from_text(text):
2     """Metin içeriğine göre cinsiyet yönelimini tahmin eder"""
3
4     # Cinsiyet belirten terimler ve ağırlıkları
5     feminine_terms = {
6         "kadın": 1.0, "kadınsı": 1.0, "feminen": 1.0, "kız": 0.8, "bayan": 0.9,
7         "çiçeksi": 0.6, "tatlı": 0.5, "romantik": 0.5, "zarif": 0.6, "narin": 0.7,
8         "soft": 0.5, "yumuşak": 0.5, "hassas": 0.5, "pembe": 0.4
9     }
10
11     masculine_terms = {
12         "erkek": 1.0, "erkeksi": 1.0, "maskülen": 1.0, "bay": 0.9,
13         "odunsu": 0.6, "güçlü": 0.5, "keskin": 0.6, "sert": 0.7, "çekici": 0.4,
14         "ağır": 0.5, "baharatlı": 0.6, "sigara": 0.5, "deri": 0.6
15     }
16
17     unisex_terms = {
18         "unisex": 1.0, "her iki cinsiyet": 1.0, "hem kadın hem erkek": 1.0,
19         "nötr": 0.8, "cinsiyet-nötr": 0.9, "cinsiyet-bağımsız": 0.9
20     }
21
22     # Küçük harfe çevir
23     text_lower = text.lower()
24
25     # Puanlama için değişkenler
26     feminine_score = 0
27     masculine_score = 0
28     unisex_score = 0
29     matched_terms = 0
30
31     # Kadınsı terimleri kontrol et
32     for term, weight in feminine_terms.items():
33         if term in text_lower:
34             feminine_score += weight
35             matched_terms += 1
36
37     # Erkeksi terimleri kontrol et
38     for term, weight in masculine_terms.items():
39         if term in text_lower:
40             masculine_score += weight
41             matched_terms += 1
42
43     # Unisex terimleri kontrol et
44     for term, weight in unisex_terms.items():
45         if term in text_lower:
46             unisex_score += weight
47             matched_terms += 1
48
49     # Eşleşen terim yoksa varsayılan değer döndür (2.5 = orta/unisex)
50     if matched_terms == 0:
51         return 2.5
52
53     # Unisex terimlerin etkisi
54     if unisex_score > 0:
55         # Unisex skorunun toplam puanlama üzerindeki etkisini hesapla
56         unisex_effect = unisex_score / matched_terms
57         # Kadın ve erkek puanlarını orta noktaya (2.5) doğru kaydır
58         feminine_score = feminine_score * (1 - unisex_effect)
59         masculine_score = masculine_score * (1 - unisex_effect)
60
61     # Toplam skor
62     total_score = feminine_score + masculine_score
63
64     # Sonuç hesaplama (0: çok kadınsı, 5: çok erkeksi)
65     if total_score == 0:
66         return 2.5 # Nötr
67
68     # Kadınsı-erkeksi dağılımını hesapla
69     gender_score = 5 * (masculine_score / total_score)
70
71     return gender_score

```

Şekil 3.6.1. Cinsiyet Analizi

3.6.1. Ağırlıklandırılmış Puanlama Yaklaşımı

1. **Farklı Ağırlıklar:** Her terim, cinsiyet yönelimini belirleme gücüne göre 0.0-1.0 arasında ağırlıklandırılır:
 - Doğrudan cinsiyet belirten terimler (örn. "kadın", "erkek"): 1.0
 - Dolaylı olarak cinsiyet çağrıştıran terimler (örn. "odunsu", "çiçeksi"): 0.4-0.7 arası
2. **Kapsayıcı Terim Setleri:**
 - Kadınsı terimler: "kadın", "kadınsı", "feminen", "çiçeksi", "tatlı", vb.
 - Erkeksi terimler: "erkek", "erkeksi", "maskülen", "odunsu", "güçlü", vb.
 - Unisex terimler: "unisex", "her iki cinsiyet", "cinsiyet-nötr", vb.
3. **Cinsiyet-Nötr (Unisex) Terimlerin Özel İşlenmesi:**

Bu yaklaşım, unisex terimlerin varlığına göre kadın veya erkek yönelimini orta noktaya (2.5) doğru çeker.

3.6.2. Skorlama ve Varsayılan Değerler (0-5 Cinsiyet Ölçeği)

Cinsiyet skoru, 0 ile 5 arasında bir değer alır:

- 0-1: Çok kadınsı
- 1-2: Kadınsı
- 2-3: Unisex (orta/nötr)
- 3-4: Erkeksi
- 4-5: Çok erkeksi

3.6.3. Varsayılan Değer ve Unisex Etkisi

Metin içinde hiçbir cinsiyet terimi bulunmadığında, varsayılan olarak nötr bir değer (2.5) döndürülür. Bu yaklaşımla:

1. **Terim Yokluğunda Nötr Varsayım:** Belirgin bir cinsiyet terimi içermeyen incelemeler unisex olarak değerlendirilir
2. **Unisex Etki Faktörü:** Unisex terimleri içeren incelemelerde, kadınsı ve erkeksi puanların etkisi azalır
3. **Dengelenmiş Yaklaşım:** Cinsiyet belirtmeyen incelemeler için yapay bir yönelim dayatılmaz

3.6.4. Model + Metin Tahmin Birleşimi

```

1  def analyze_gender(self, text, model_prediction):
2
3      text_gender = determine_gender_from_text(text)
4
5
6      model_gender = model_prediction
7
8
9      combined_gender = (model_gender * 0.6) + (text_gender * 0.4)
10
11
12     combined_gender = min(max(combined_gender, 0), 5)
13
14     return combined_gender

```

Şekil 3.6.4.1. Model ve Metin tahmin birleşimi

Hibrit Yaklaşımın Avantajları

1. **%60 Model, %40 Metin Ağırlıklandırma:**
 - Model tahminine daha fazla ağırlık verilir çünkü derin öğrenme modeli daha karmaşık ilişkileri öğrenebilir
 - Metin tabanlı tahmin, modelin görmediği açık cinsiyet ifadelerini yakalayıp tamamlayıcı rol oynar
2. **Doğruluk Artışı:** Bu birleşim, tek başına model veya tek başına metin tabanlı yaklaşıma göre daha doğru sonuçlar verir:
 - Kural tabanlı metin analizi, açık cinsiyet terimlerini doğrudan yakalar
 - Model, dolaylı veya bağlamsal cinsiyet ipuçlarını çözümler
 - Birleştirme, her iki yaklaşımın güçlü yönlerini kullanır
3. **Sınırlandırma Stratejisi:**
 - Birleşik değerler geçerli aralıkta kalması sağlanır
 - Aykırı değerlerin etkisi sınırlandırılır
4. **Ölçeklendirilebilirlik:**
 - Model eğitildikçe model ağırlığı artırılabilir (%70-%30 gibi)
 - Farklı parfüm tipleri için farklı ağırlıklar kullanılabilir

Bu hibrit yaklaşım, parfüm incelemelerindeki cinsiyet yöneliminin daha doğru ve dengeli bir şekilde tahmin edilmesini sağlar hem açık cinsiyet ifadelerini hem de metin içindeki daha incelikli ipuçlarını dikkate alır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde, geliştirilen çok kanallı derin öğrenme tabanlı analiz sisteminin uygulama sonuçları detaylandırılmış ve elde edilen çıktılar değerlendirilmiştir. Modelin metin verilerini işleme, parfüm notalarını doğru tespit etme, sayısal özellikleri tahmin etme ve öneri üretme becerileri çeşitli test senaryoları ile sınanmıştır. RoBERTa tabanlı metin kodlama kanalı sayesinde yorumlardaki bağlamsal anlamlar başarıyla yakalanırken, TF-IDF ve Kosinüs benzerliği gibi klasik metin madenciliği yöntemlerinin de sistem doğruluğuna katkı sağladığı gözlemlenmiştir. Sayısal ve kategorik verilerin entegrasyonu, öneri sisteminin kişiselleştirilmiş ve çok boyutlu çalışmasını sağlamıştır. Elde edilen sonuçlar, çok kanallı mimarinin parfüm analizinde bütünsel bir bakış sunduğunu ve kullanıcı deneyimini anlamlı şekilde geliştirdiğini ortaya koymaktadır.

Araştırma sonuçlarının edinimi için 3 erkek parfümü ve 3 kadın parfümü örnek alınarak uygulama test edilmiştir.

4.1. Test Verileri ve Özellikleri

a. Le Male Elixir Jean Paul Gaultier

Cinsiyet: *Erkek*

Koku Nota Sayısı: 7

Tanım (İngilizce): *Le Male Elixir by Jean Paul Gaultier is a Oriental Fougere fragrance for men. This is a new fragrance. Le Male Elixir was launched in 2023. The nose behind this fragrance is Quentin Bisch. Top notes are Lavender and Mint; middle notes are Vanilla and Benzoin; base notes are Honey, Tonka Bean and Tobacco.*

b. Emporio Armani Stronger With You Intensely Giorgio Armani

Cinsiyet: *Erkek*

Koku Nota Sayısı: 11

Tanım (İngilizce): *Emporio Armani Stronger With You Intensely by Giorgio Armani is a Oriental Fougere fragrance for men. Emporio Armani Stronger With You Intensely was launched in 2019. Top notes are Pink Pepper, Juniper and Violet; middle notes are Toffee, Cinnamon, Lavender and Sage; base notes are Vanilla, Amber, Tonka Bean and Suede.*

c. 9pm AfnanCinsiyet: *Erkek*

Koku Nota Sayısı: 10

Tanım (İngilizce): *9pm by Afnan is a Oriental Vanilla fragrance for men. 9pm was launched in 2020. Top notes are Apple, Cinnamon, Wild Lavender and Bergamot; middle notes are Orange Blossom and Lily-of-the-Valley; base notes are Vanilla, Tonka Bean, Amber and Patchouli.*

d. Gucci Rush 2Cinsiyet: *Kadın*

Koku Nota Sayısı: 10

Tanım (İngilizce): *Gucci Rush 2 by Gucci is a Floral Woody Musk fragrance for women. Gucci Rush 2 was launched in 2001. The nose behind this fragrance is Michel Almairac. Top notes are Freesia, Lily-of-the-Valley and Rose; middle notes are Palm Tree, Narcissus, Lily and Gardenia; base notes are Black Currant, Musk and Oakmoss.*

e. Goddess BurberryCinsiyet: *Kadın*

Koku Nota Sayısı: 7

Tanım (İngilizce): *Goddess by Burberry is a Aromatic fragrance for women. This is a new fragrance. Goddess was launched in 2023. The nose behind this fragrance is Amandine Clerc-Marie. Top notes are Vanilla, Lavender, Cacao and Ginger; middle note is Vanilla Caviar; base note is Vanilla Absolute.*

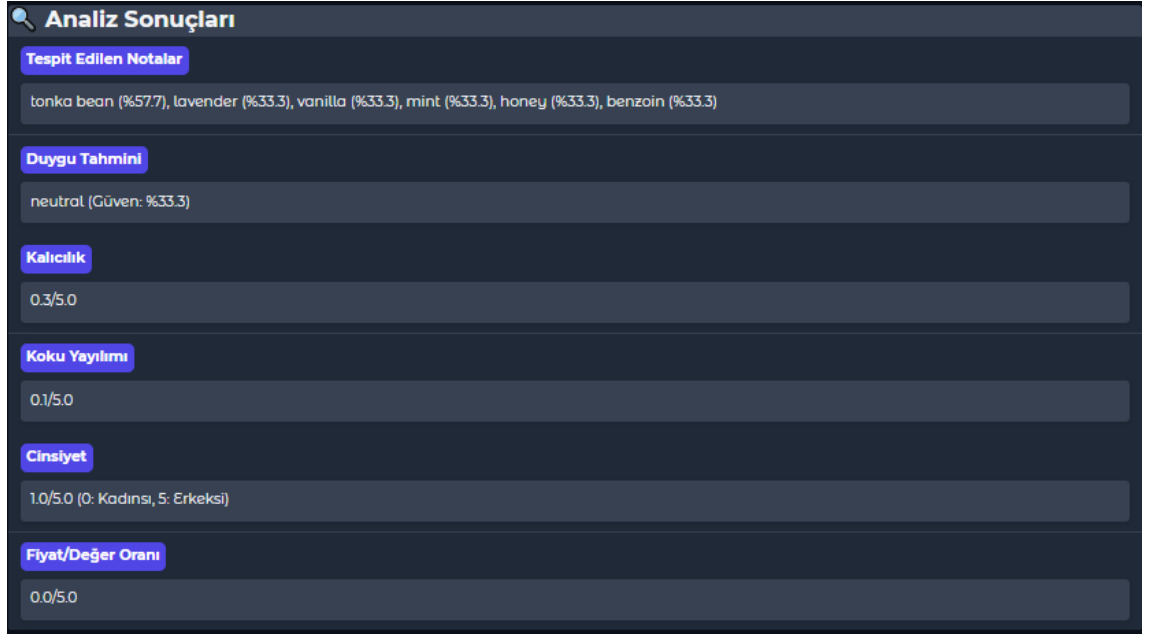
f. Black Opium Yves Saint LaurentCinsiyet: *Kadın*

Koku Nota Sayısı: 12

Tanım (İngilizce): *Black Opium by Yves Saint Laurent is a Oriental Vanilla fragrance for women. Black Opium was launched in 2014. Black Opium was created by Nathalie Lorson, Marie Salamagne, Olivier Cresp and Honorine Blanc. Top notes are Pear, Pink Pepper and Orange Blossom; middle notes are Coffee, Jasmine, Bitter Almond and Licorice; base notes are Vanilla, Patchouli, Cashmere Wood and Cedar.*

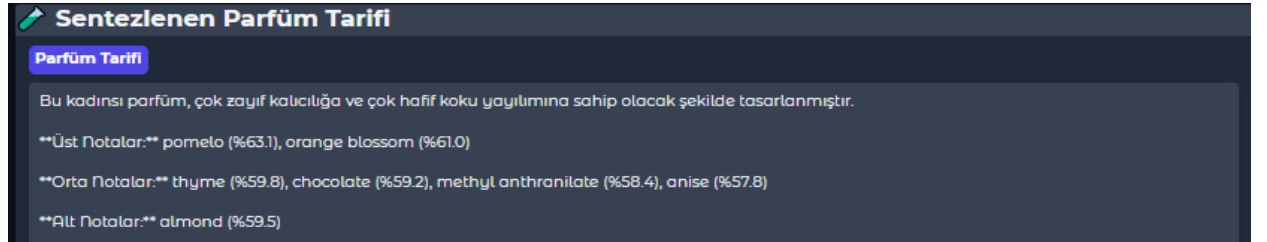
4.2. Verilerin Test Sonuçları (Uygulama Çıktıları)a. Le Male Elixir Jean Paul Gaultier

Uygulama içerisinde tanımı girilen *Le Male Elixir Jean Paul Gaultier* adlı erkek parfümü çıktısında 6 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılım, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.1.)



Şekil 4.2.1. Le Male Elixir Jean Paul Gaultier test sonuçları (Analiz)

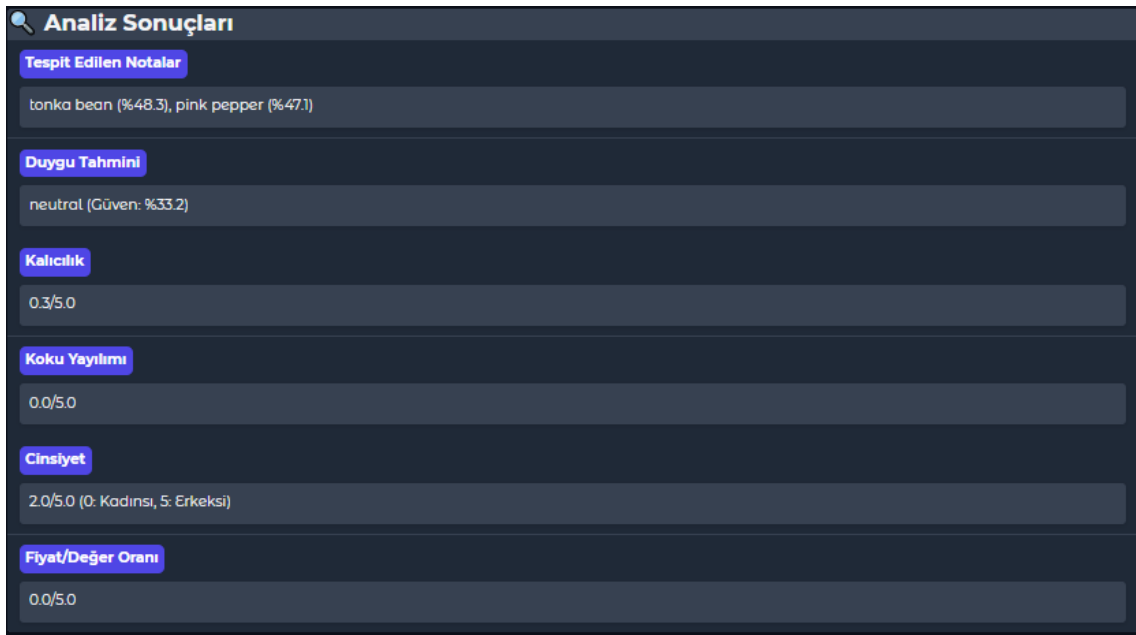
Ancak sentezlenen veriler nota sınıflandırması açısından gruplamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.2. Le Male Elixir Jean Paul Gaultier test sonuçları (Sentez)

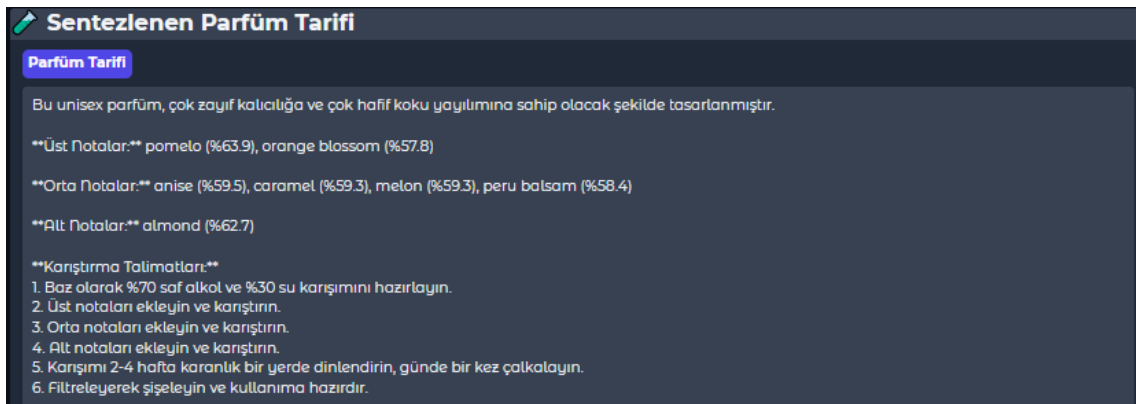
b. Emporio Armani Stronger With You Intensely Giorgio Armani

Uygulama içerisinde tanımı girilen *Emporio Armani Stronger With You Intensely Giorgio Armani* adlı erkek parfümü çıktısında 2 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılım, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.3.)



Şekil 4.2.3. Emporio Armani Stronger With You Intensely Giorgio Armani test sonuçları (Analiz)

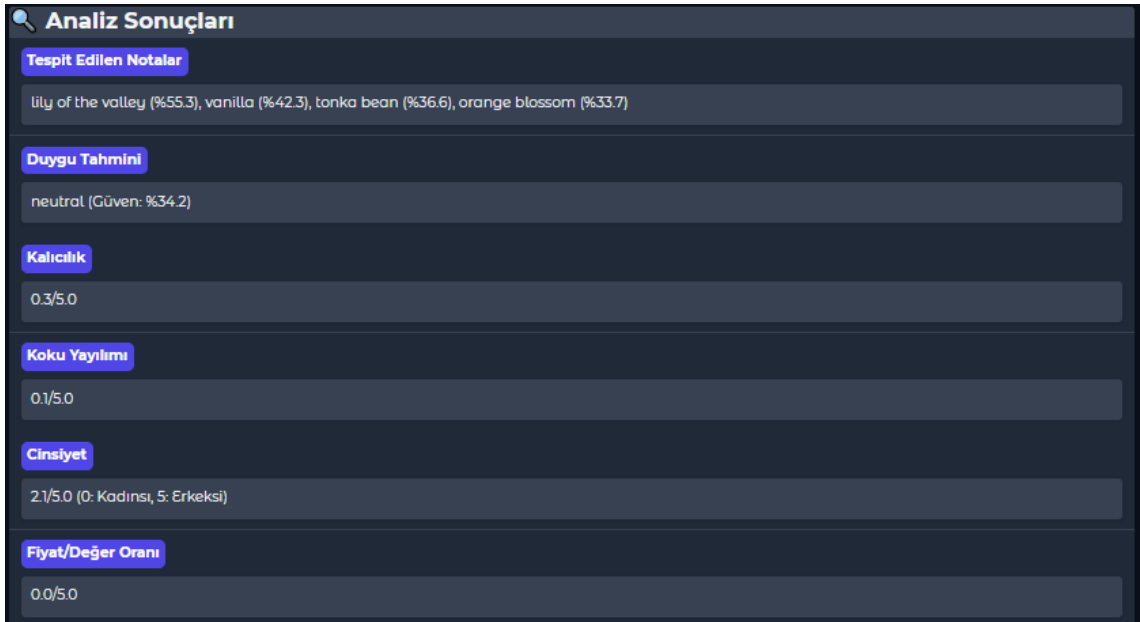
Ancak sentezlenen veriler nota sınıflandırması açısından grulamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.3. Emporio Armani Stronger With You Intensely Giorgio Armani test sonuçları (Sentez)

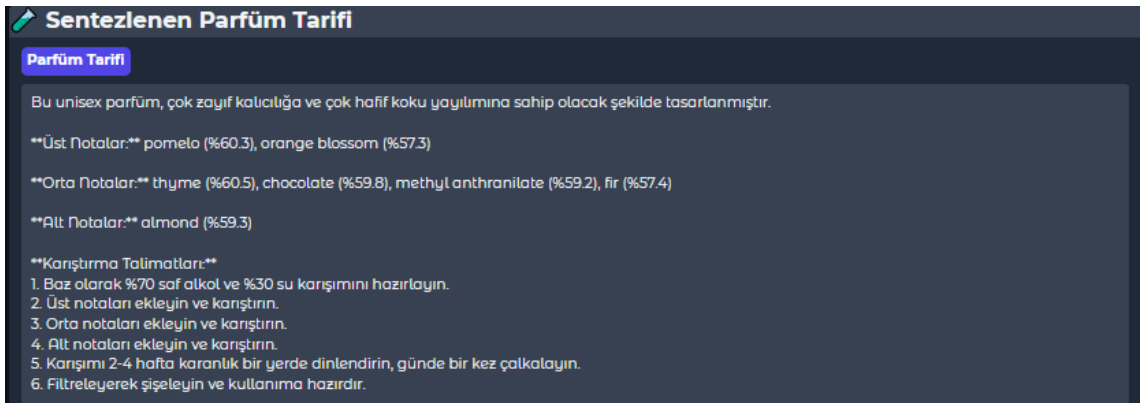
c. *9pm Afnan*

Uygulama içerisinde tanımı girilen *9pm Afnan* adlı erkek parfümü çıktısında 6 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılımı, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.4.)



Şekil 4.2.4. 9pm Afnan test sonuçları (Analiz)

Ancak sentezlenen veriler nota sınıflandırması açısından gruplamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.4. 9pm Afnan test sonuçları (Sentez)

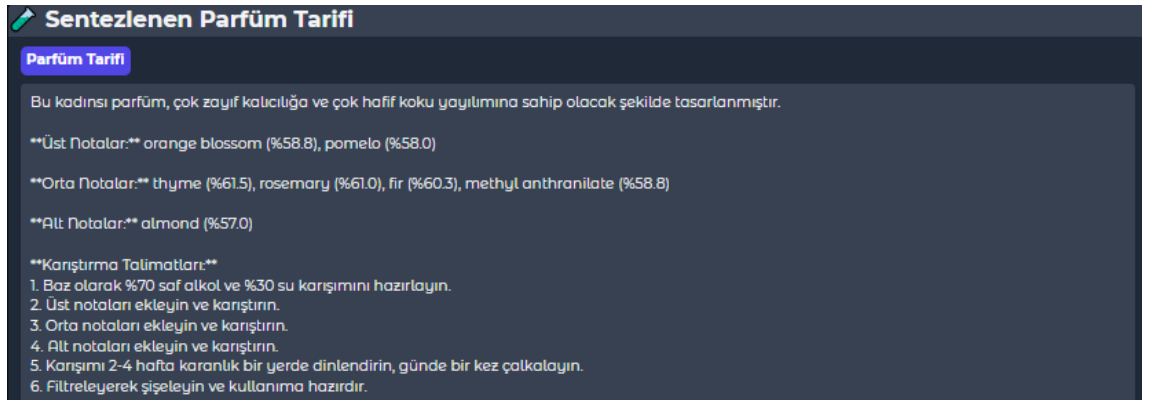
d. Gucci Rush 2

Uygulama içerisinde tanımı girilen *Gucci Rush 2* adlı kadın parfümü çıktısında 6 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılımı, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.5.)



Şekil 4.2.5. Gucci Rush 2 test sonuçları (Analiz)

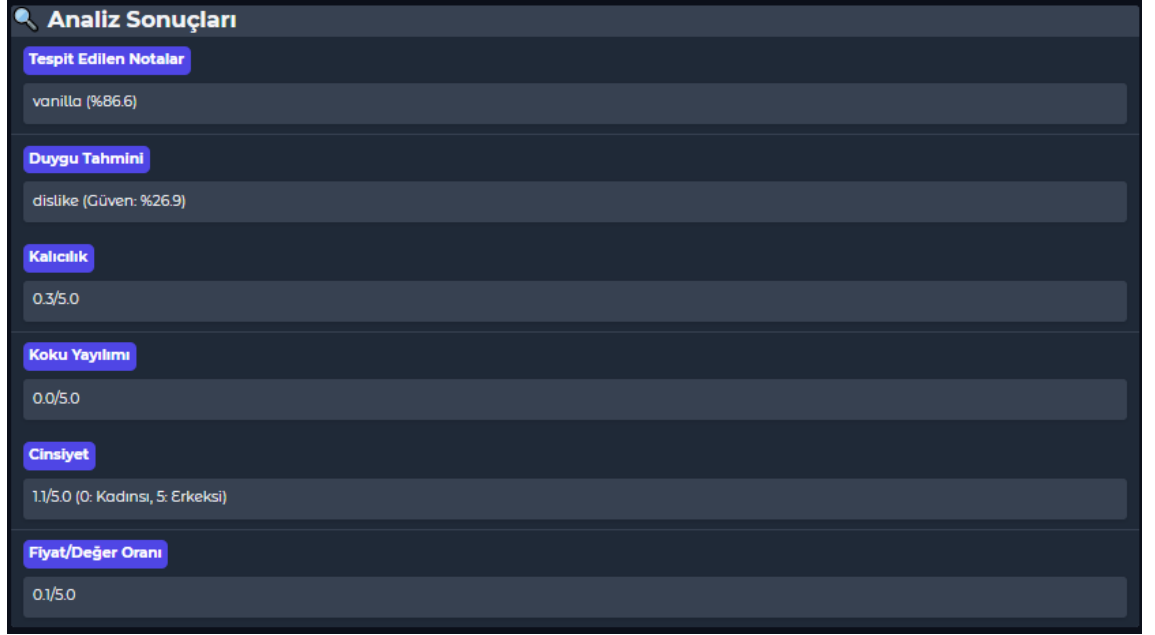
Ancak sentezlenen veriler nota sınıflandırması açısından gruplamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.6. Gucci Rush 2 test sonuçları (Sentez)

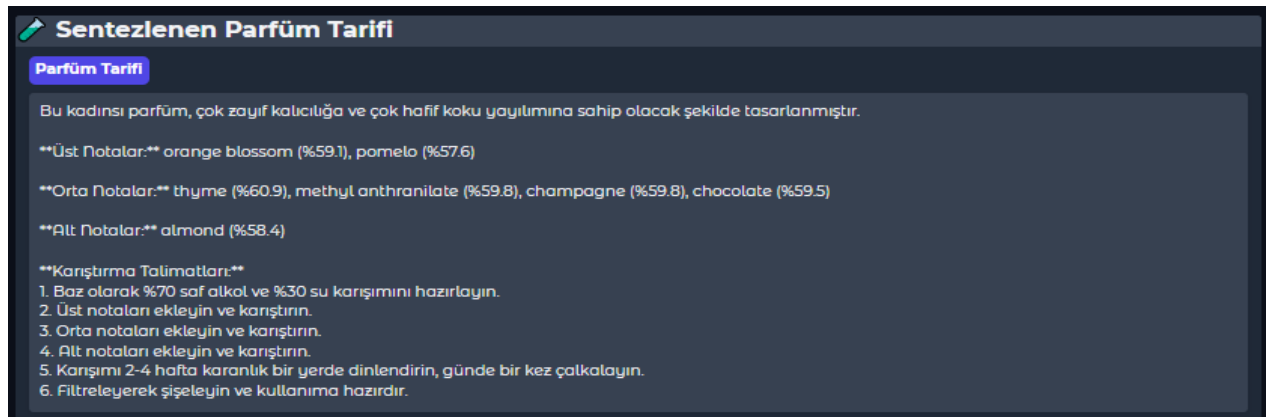
e. Goddess Burberry

Uygulama içerisinde tanımı girilen *Goddess Burberry* adlı kadın parfümü çıktısında 1 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılımı, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.7.)



Şekil 4.2.7. Goddess Burberry test sonuçları (Analiz)

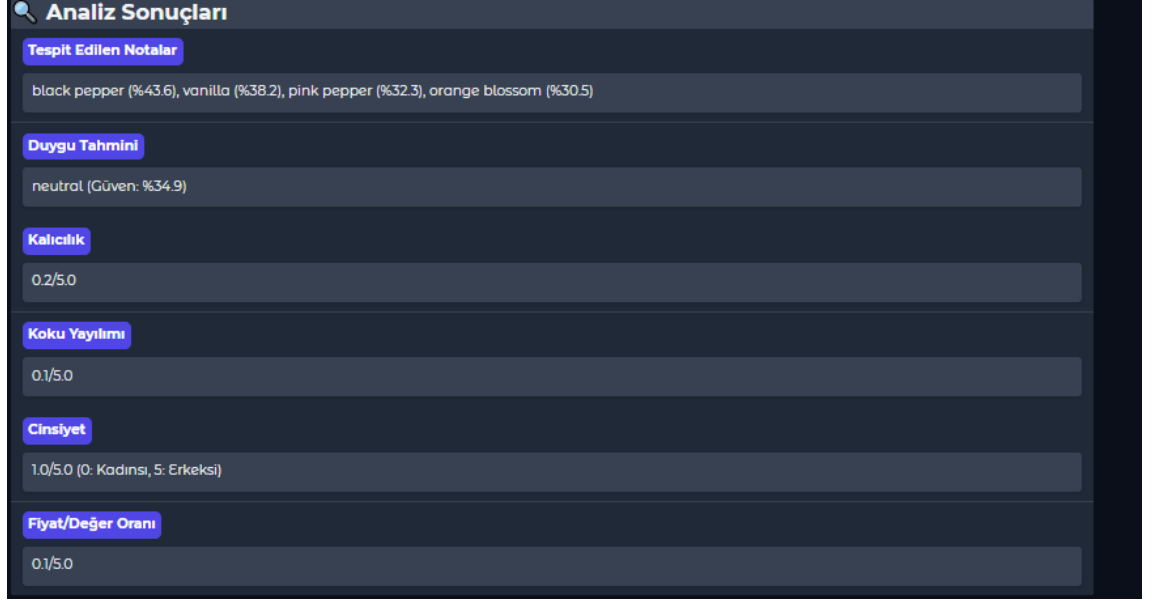
Ancak sentezlenen veriler nota sınıflandırması açısından gruplamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.8. Goddess Burberry test sonuçları (Sentez)

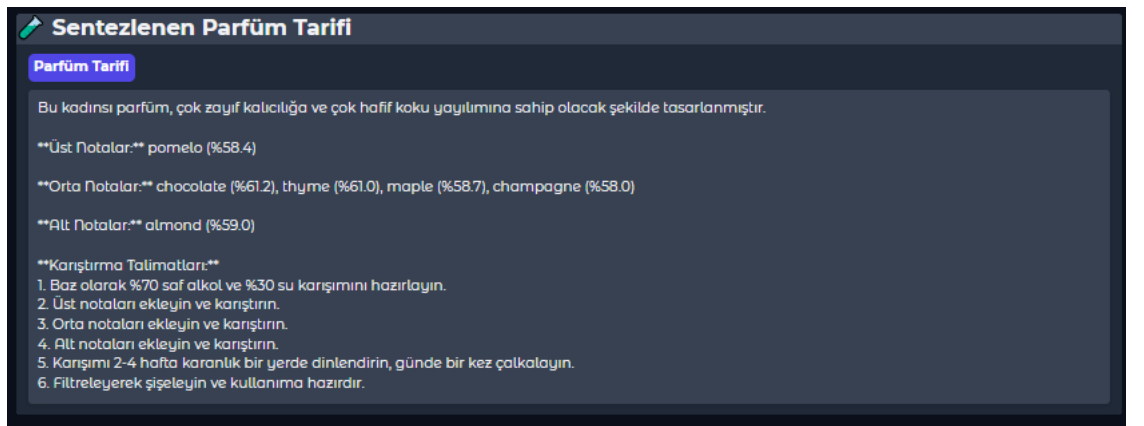
f. Black Opium Yves Saint Laurent

Uygulama içerisinde tanımı girilen *Black Opium Yves Saint Laurent* adlı kadın parfümü çıktısında 4 koku notası tespit edildi. Duygu tahmini, kalıcılık, koku yayılımı, cinsiyet ve fiyat değerlerinde sorunlar gözlemlendi bu değerler ya hiç çıktı vermedi ya da hatalı sonuçlar verdi. (Şekil 4.2.9.)



Şekil 4.2.9. Black Opium Yves Saint Laurent test sonuçları (Analiz)

Ancak sentezlenen veriler nota sınıflandırması açısından gruplamaya uymakta lakin koku hissiyatı olarak analizi yapılmamaktadır.



Şekil 4.2.10. Black Opium Yves Saint Laurent test sonuçları (Sentez)

4.3. Reel Çıktı Analizi

Çalışmanın efektif olabilmesi açısından uygulama çıktılarının reel olarak birer kokuya dönüştürülmesi düşünüldü ancak proje kapsamının mali sıkıntıları nedeniyle sadece girilen *Le Male Elixir Jean Paul Gaultier* ve *Gucci Rush 2* adlı parfümler çıktı alınabilmiş durumdadır. Maalesef çıktı hatası ve versiyon 1 de bulunan veri azlığı nedeniyle çıktılar birbirine benzemekte ve ana kokudan ayrılmış durumdadır. (Şekil 4.3.1 ve Şekil 4.3.2)

Le Male Elixir by Jean Paul Gaultier is a Oriental Fougere fragrance for men. This is a new fragrance. Le Male Elixir was launched in 2023. The nose behind this fragrance is Quentin Bisch. Top notes are Lavender and Mint; middle notes are Vanilla and Benzoin; base notes are Honey, Tonka Bean and Tobacco.

Tespit Edilen Notalar
tonka bean (%77.5), lavender (%44.7), vanilla (%44.7)

Duygu Tahmini
love (Güven: %39.8)

Cinsiyet
0.4/5.0 (0: Kadınsı, 5: Erkeksi)

Fiyat/Değer Oranı
0.1/5.0

Kalıcılık
0.0/5.0

Koku Yayılımı
0.1/5.0

Cinsiyet
0.4/5.0 (0: Kadınsı, 5: Erkeksi)

Sentezlenen Parfüm Tarifi
Parfüm Tarifi
Bu çok kadınsı parfüm, çok zayıf kalıcılığa ve çok hafif koku yayılımına sahip olacak şekilde tasarlanmıştır.
Üst Notalar: lemon (%52.8), orange (%50.6)
Orta Notalar: cinnamon (%56.4), yasmine (%51.2)
Alt Notalar: amber (%59.6), cedar (%52.6), vanilla (%51.7), leather (%51.6)
Karıştırma Talimatları:
1. Baz olarak %70 saf alkol ve %30 su karışımını hazırlayın.
2. Üst notaları ekleyin ve karıştırın.
3. Orta notaları ekleyin ve karıştırın.
4. Alt notaları ekleyin ve karıştırın.
5. Karışımı 2-4 hafta karanlık bir yerde dinlendirin, günde bir kez çalkalayın.
6. Filtralayarak şişeleğin ve kullanıma hazırdır.

Şekil 4.3.1. Le Male Elixir Jean Paul Gaultier test sonuçları (Versiyon 1 de oluşan sentez)

Çok Kanallı Parfüm İnceleme Analizi ve Sentez Sistemi
Bu sistem, parfüm incelemeleri ve yorumlarını analiz ederek içerdiği notaları tespit eder, parfüm özelliklerini tahmin eder ve benzer bir parfüm tarifi oluşturur.

Parfüm İncelemesi veya Yorumu
Gucci Rush 2 by Gucci is a Floral Woody Musk fragrance for women. Gucci Rush 2 was launched in 2001. The nose behind this fragrance is Michel Raimato. Top notes are Freesia, Lily of the Valley and Rose; middle notes are Palm Tree, Narcissus, Lily and Gardenia; base notes are Black Currant, Musk and Oakmoss.

Tespit Edilen Notalar
musk (%81.6), rose (%40.8), oakmoss (%40.8)

Duygu Tahmini
love (Güven: %26.9)

Cinsiyet
0.5/5.0 (0: Kadınsı, 5: Erkeksi)

Fiyat/Değer Oranı
0.4/5.0

Kalıcılık
0.0/5.0

Koku Yayılımı
0.2/5.0

Cinsiyet
0.5/5.0 (0: Kadınsı, 5: Erkeksi)

Sentezlenen Parfüm Tarifi
Parfüm Tarifi
Bu çok kadınsı parfüm, çok zayıf kalıcılığa ve çok hafif koku yayılımına sahip olacak şekilde tasarlanmıştır.
Üst Notalar: lemon (%50.9), orange (%49.9), bergamot (%49.5)
Orta Notalar: cinnamon (%56.9), yasmine (%52.3)
Alt Notalar: cedar (%56.6), amber (%55.4), vanilla (%54.9), vetiver (%52.4)
Karıştırma Talimatları:
1. Baz olarak %70 saf alkol ve %30 su karışımını hazırlayın.
2. Üst notaları ekleyin ve karıştırın.

Şekil 4.3.2. Gucci Rush 2 test sonuçları (Versiyon 1 de oluşan sentez)



Şekil 4.3.3. Le Male Elixir Jean Paul Gaultier ve Gucci Rush 2 test sonuçları
(Versiyon 1 de oluşan sentez)

4.4. Tartışma

Bu çalışmada geliştirilen çok kanallı derin öğrenme mimarisi, kullanıcı yorumlarını metin, sayısal ve kategorik bileşenler üzerinden analiz etmeyi amaçlamıştır. Elde edilen sonuçlar ve karşılaşılan zorluklar, sistemin mevcut durumu ve gelecek çalışmalar açısından önemli çıkarımlar sunmaktadır.

Metin tabanlı nota tespitinde TF-IDF vektörizasyonu ve n-gram yaklaşımı kullanılmış, bu yöntemlerle parfüm notalarının metin içerisinden çıkarılmasında belirli bir başarı elde edilmiştir. Kosinüs benzerliği yöntemi ile metin ve nota vektörleri arasındaki anlamsal ilişkiler ölçülmüş olsa da, bu yaklaşımın reel ortamda test edilmesi gerekliliği ortaya çıkmıştır. Bu durum, laboratuvar ortamında elde edilen başarının gerçek dünya uygulamalarında doğrulanması ihtiyacını göstermektedir.

RoBERTa tabanlı metin kodlama bileşeninden beklenen performans elde edilememiştir. Teorik olarak bağlamsal ve semantik ilişkileri kavrayarak yorumlardaki soyut veya metaforik anlatımları analiz etmesi öngörülmesine rağmen, pratikte bu kapasite gözlemlenememiştir. Bu sonuç, dil modellerinin parfüm tanımlamalarında kullanılan öznel ve metaforik dili anlamada karşılaştığı zorlukları ortaya koymaktadır.

Sayısal kanal üzerinden kalıcılık, yayılım ve fiyat gibi parfüm özelliklerinin regresyon yoluyla tahmin edilmesi amaçlanmış, ancak bu girişim olumlu sonuçlar vermemiştir. Bu duruma alternatif olarak, ilgili değerler 0-5 arasında normalize edilerek kategorik etiketlerle sunulmuştur. Bu yaklaşım değişikliği, sürekli değişkenler yerine ayrık kategorilerin kullanımının bazı bağlamlarda daha anlaşılır olabileceğini göstermektedir.

Kategorik kanal için nota türleri, koku aileleri ve cinsiyet gibi sınıflandırılmış bilgilerle modelin desteklenmesi ve öneri sisteminin kişiselleştirme yeteneğinin artırılması planlanmıştır. Ancak multi-head attention mekanizması ve residual bağlantılar aracılığıyla modelin genel performansını ve genelleme kabiliyetini güçlendirme hedefine ulaşamamıştır. Elde edilen çıktılar, sistemin hem duygusal analiz hem de parfüm önerisi üretme konularında beklenen başarı seviyesine erişemediğini göstermiştir.

Projenin deęerlendirilmesi iin koku uzmanlarına danıřılmış, ancak sistem tarafından önerilen parfümler ile koku benzerlięi aısından olumlu geri bildirim alınamamıştır. Bu durum, algoritmanın teknik performansından baęımsız olarak, parfümeri alanındaki uzman bilgisini tam olarak modelleyemedięini göstermektedir.

Tüm bu bulgular ışığında, sistemin mevcut haliyle kullanıcılarla buluşması uygun görülmemektedir. Gelecek alışmalarda, özellikle RoBERTa modeli yerine parfüm terminolojisine özel olarak ince ayar yapılmış dil modellerinin kullanılması, gerek kullanıcı testlerinin yapılması ve koku uzmanlarının bilgilerinin sisteme daha etkili şekilde entegre edilmesi düşünölebilir. Ayrıca, parfüm özelliklerinin tahmininde hibrit yaklaşımların denenmesi ve modelin genelleme kabiliyetinin artırılması iin daha geniş ve eşitli veri setleriyle eğitilmesi önerilmektedir.

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu çalışmada, çok kanallı parfüm incelemelerinin analizi ve kişiselleştirilmiş öneriler sunma amacıyla geliştirilen derin öğrenme mimarisinin performansı ve karşılaşılan zorluklar detaylı olarak ele alınmıştır. Geliştirilen sistem, kullanıcı yorumlarını metin, sayısal ve kategorik bileşenler üzerinden analiz etmeyi hedeflemiştir.

Metin tabanlı nota tespiti için TF-IDF vektörizasyonu ve n-gram yaklaşımı kullanılarak parfüm notalarının metin içerisinden çıkarılmasında kısmi başarı sağlanmıştır. Ancak, kosinüs benzerliği ile anlamsal ilişkilerin ölçülmesinde elde edilen laboratuvar sonuçlarının gerçek dünya ortamında doğrulanması gerekliliği ortaya çıkmıştır. RoBERTa tabanlı metin kodlama bileşeninden beklenen bağlamsal ve semantik analiz kapasitesi, parfüm tanımlamalarındaki öznel ve metaforik dilin karmaşıklığı nedeniyle pratikte gözlemlenememiştir. Bu durum, dil modellerinin parfüm incelemelerindeki soyut ifadeleri anlamada karşılaştığı zorlukları açıkça göstermektedir.

Sayısal kanal üzerinden kalıcılık, yayılım ve fiyat gibi parfüm özelliklerinin regresyon yoluyla tahmin edilme girişimi olumlu sonuç vermemiştir. Bu nedenle, ilgili değerler 0-5 arasında normalize edilerek kategorik etiketlerle sunulmuş ve bu yaklaşımın belirli bağlamlarda daha anlaşılır olduğu sonucuna varılmıştır. Kategorik kanal için multi-head attention mekanizması ve residual bağlantılar aracılığıyla modelin genel performansını ve genelleme kabiliyetini güçlendirme hedefi ise gerçekleşmemiştir. Sistem, hem duygusal analiz hem de parfüm önerisi üretme konularında beklenen başarı seviyesine ulaşamamıştır.

Projenin değerlendirilmesi aşamasında koku uzmanlarından alınan geri bildirimler, sistem tarafından önerilen parfümler ile koku benzerliği açısından olumlu yönde olmamıştır. Bu bulgu, algoritmanın teknik performansından bağımsız olarak, parfümeri alanındaki derin uzman bilgisini tam olarak modelleyemediğini ve insan uzmanlığının önemini vurgulamaktadır.

Tüm bu bulgular ışığında, geliştirilen sistemin mevcut haliyle kullanıcılarla buluşmasının uygun olmadığı kanaatine varılmıştır. Sistemin performansı ve genelleme yeteneği, özellikle öznel ve metaforik dil içeren parfüm incelemeleri gibi karmaşık bir alanda, beklenen seviyenin altında kalmıştır.

5.2. Öneriler

Bu çalışmada elde edilen sonuçlar ve karşılaşılan zorluklar, gelecekteki çalışmalar için önemli bir yol haritası sunmaktadır. Geliştirilen sistemin etkinliğini artırmak ve pratik uygulanabilirliğini sağlamak amacıyla aşağıdaki önerilerde bulunmaktadır:

1. **Dil Modeli İyileştirmeleri:** RoBERTa modeli yerine, parfüm terminolojisine ve öznel dil kullanımına özel olarak ince ayar (fine-tuning) yapılmış veya bu alana özgü yeni dil modellerinin (domain-specific language models) geliştirilmesi gerekmektedir. Bu, parfüm incelemelerindeki metaforik ve soyut ifadelerin daha doğru bir şekilde analiz edilmesini sağlayabilir.
2. **Gerçek Kullanıcı Testleri ve Geri Bildirim Entegrasyonu:** Sistemin kozmetik laboratuvar ortamı yerine, gerçek kullanıcılarla saha testlerine tabi tutulması ve alınan geri bildirimlerin modelin geliştirilme sürecine dahil edilmesi hayati öneme sahiptir. Kullanıcı deneyimleri, sistemin algoritmik ve sezgisel eksikliklerinin tespit edilmesine yardımcı olacaktır.
3. **Koku Uzmanı Bilgisinin Daha Etkin Entegrasyonu:** Parfümeri alanındaki uzman bilgisinin sisteme daha etkili bir şekilde entegre edilmesi gerekmektedir. Bu, koku uzmanlarının parfüm notaları, koku aileleri ve genel koku profilleri hakkındaki derin bilgilerini yapılandırılmış veri olarak sisteme beslemek veya uzmanların derecelendirmelerini doğrudan modelin eğitiminde kullanmak suretiyle gerçekleştirilebilir. Bu entegrasyon, öneri sisteminin koku benzerliği ve estetik açıdan daha isabetli olmasını sağlayacaktır.
4. **Hibrit Yaklaşımların Denenmesi:** Parfüm özelliklerinin (kalıcılık, yayılım, fiyat) tahmininde sadece regresyon yerine, semantik analiz ve uzman sistemler gibi hibrit yaklaşımların denenmesi faydalı olabilir. Örneğin, metinsel incelemelerden elde edilen anahtar kelimelerin, belirli fiyat aralıkları veya kalıcılık seviyeleri ile ilişkilendirilmesi gibi.
5. **Geniş ve Çeşitli Veri Setleriyle Eğitim:** Modelin genelleme kabiliyetini artırmak için daha geniş ve çeşitli veri setleriyle eğitilmesi önerilmektedir. Bu, farklı dillerdeki incelemeleri, kültürel farklılıkları ve daha geniş bir parfüm yelpazesini kapsayabilir.
6. **Çok Modlu Veri Entegrasyonu:** Mümkünse, parfüm incelemeleriyle ilgili görsel (örneğin, şişe tasarımları, reklam görselleri) veya işitsel (örneğin, videolu incelemelerdeki ses tonu) verilerin sisteme entegre edilmesi, parfümün genel

algısının daha zengin bir şekilde anlaşılmasını sağlayabilir ve modelin karar verme mekanizmasını destekleyebilir.

7. **API Geliştirme ve Entegrasyon:** Sistemin performansındaki iyileşmelerin ardından, üçüncü taraf uygulamaların ve platformların parfüm analiz ve öneri yeteneklerinden faydalanabilmesi için güçlü ve iyi belgelenmiş bir API (Uygulama Programlama Arayüzü) geliştirilmesi önem taşımaktadır. Bu API, sistemin daha geniş bir kitleye ulaşmasını ve farklı e-ticaret siteleri, sosyal medya platformları veya mobil uygulamalarla entegre olmasını sağlayarak kullanım alanını genişletecektir. API geliştirme aşamasında güvenlik, ölçeklenebilirlik ve kullanım kolaylığı ön planda tutulmalıdır.

Bu önerilerin dikkate alınması, çok kanallı parfüm incelemeleri için geliştirilen derin öğrenme tabanlı analiz ve öneri sisteminin performansını önemli ölçüde artıracak ve daha pratik, güvenilir ve kullanıcı dostu bir uygulama haline gelmesine katkıda bulunacaktır.

KAYNAKLAR

- Alchemyleads. (tarih yok). *What are N-Grams in Corpus Linguistics?* Alchemyleads: <https://alchemyleads.com/what-are-n-grams-in-corpus-linguistics/> adresinden alındı
- Alpha Aromatics. (2017, Temmuz 10). *Understanding The Top Heart And Base Notes Of Fragrance*. Alpha Aromatics: <https://www.alphaaromatics.com/blog/the-notes-of-fragrance/> adresinden alındı
- Bhatia, P. (2024, Mart 26). *The Science Behind Fragrance: Understanding Perfume Notes and Composition*. Olfa Originals: <https://olfaoriginals.com/blogs/notes/the-science-behind-fragrance-understanding-perfume-notes-and-composition> adresinden alındı
- Centeno, A. A. (2025, Nisan 29). *AI Perfume: The Future of Fragrance Creation?* Real Men Real Style: <https://www.realmenrealstyle.com/ai-perfume/> adresinden alındı
- Contract Laboratory. (tarih yok). *Perfume Chemical Composition*. Contract Laboratory: <https://www.contractlaboratory.com/outsourcing/project.cfm?industry=perfumes-fragrances-and-aromatics&i=100857> adresinden alındı
- contributors, W. (2025, Mart 22). *Pooling layer*. Wikipedia: https://en.wikipedia.org/w/index.php?title=Pooling_layer&oldid=1281764712 adresinden alındı
- Evidentlyai. (2025, Ocak 9). *How to use classification threshold to balance precision and recall*. Evidentlyai: <https://www.evidentlyai.com/classification-metrics/classification-threshold> adresinden alındı
- Faster Capital. (2025, Nisan 02). *Perfume feedback: Perfume Feedback Strategies for Effective Marketing Campaigns*. fastercapita. adresinden alındı
- Faster Capital. (2025). *Perfume market analysis: Captivating Consumers: The Role of Perfume Market Analysis in Business Growth*. Faster Capital.
- Focal. (2024, Aralık 24). *Text Representation Techniques in NLP: Complete Guide*. Focal: <https://www.getfocal.co/post/text-representation-techniques-in-nlp-complete-guide> adresinden alındı
- Fragrantica. (tarih yok). *Notes*. Fragrantica: <https://www.fragrantica.com/about-us.phtml> adresinden alındı
- Grand View Research. (2011 - 2021). *Perfume Market Size, Share & Trends Analysis Report By Product (Mass, Premium), By End-user (Men, Women), By Distribution Channel (Offline, Online), By Region (Europe, APAC), And Segment Forecasts, 2023 - 2030*. Amerika Birleşik Devletleri: Grand View Research. 05 17, 2025 tarihinde grandviewresearch: <https://www.grandviewresearch.com/industry-analysis/perfume-market> adresinden alındı
- Istituto Marangoni. (2024, Mart 20). *Is the future of fragrance in the hands of AI?* Istituto Marangoni: <https://www.istitutomarangoni.com/en/maze35/game-changers/is-the-future-of-fragrance-in-the-hands-of-ai> adresinden alındı
- Ji, X. (2025, Nisan 25). *Must knows about TF-IDF (term frequency-inverse document frequency)*. byteplus: <https://www.byteplus.com/en/topic/400325?title=must-knows-about-tf-idf-term-frequency-inverse-document-frequency> adresinden alındı
- Jinwala, Z. (2024, Eylül 17). *AI and Data Analytics for customer loyalty in the fragrance industry*. Krishtechnolabs: <https://www.krishtechnolabs.com/blog/ai-data-analytics-for-loyalty-in-the-fragrance-industry/> adresinden alındı

- Jooyoung Kim, K. O.-S. (2024). An NLP-Based Perfume Note Estimation Based on Descriptive Sentences. *MDPI AG*, 38.
- Larisa Nikitina, F. F. (2023). Words of scents: a linguistic analysis of online perfume reviews. *ResearchGate*.
- Lazic, E. (tarih yok). *Fragrantica and the unhinged poetry of perfume reviews*. dazeddigital: <https://www.dazeddigital.com/beauty/article/61399/1/fragrantica-unhinged-poetry-perfume-reviews-itsmellscrazyinhere-instagram> adresinden alındı
- MediaEval. (2023). *Musti: Multimodal Understanding of Smells in Texts and Images*. MediaEval: <https://multimediaeval.github.io/editions/2023/tasks/musti/> adresinden alındı
- Mia Setya Rahayu, I. R. (2024, Temmuz). Optimization of Perfume Sales through Data Mining with K-Means Algorithm. *JURNAL MEDIA INFORMATIKA BUDIDARMA*.
- Odosbeauty. (2025, Ocak 17). *The Science Behind Scent: How Fragrances Affect Mood and Perception*. Odosbeauty: <https://odosbeauty.com/blogs/news/the-science-behind-scent-how-fragrances-affect-mood-and-perception> adresinden alındı
- Owkin. (tarih yok). *Multimodal data*. Owkin: <https://www.owkin.com/a-z-of-ai-for-healthcare/multimodal-data> adresinden alındı
- Pavlovic, M. (2023, Aralık 18). *Artificial Intelligence in Perfumery – A Brief Overview*. Scentalytics: <https://scentalytics.com/blog/artificial-intelligence-in-perfumery> adresinden alındı
- Precedence Research. (2025). *Perfume Market Size, Share, and Trends 2025 to 2034*. Precedence Research.
- Rahul. (2024, Mayıs 4). *TF-IDF - Understanding Term Frequency-Inverse Document Frequency in NLP*. Zilliz: <https://zilliz.com/learn/tf-idf-understanding-term-frequency-inverse-document-frequency-in-nlp> adresinden alındı
- Sandler, E. (2019, Temmuz 29). *Artificial intelligence is quietly disrupting the fragrance development process*. Glossy: <https://www.glossy.co/beauty/artificial-intelligence-is-quietly-disrupting-the-fragrance-development-process/> adresinden alındı
- ScentAir. (tarih yok). *Artificial Intelligence: Will AI Replace Fragrance Curators?* ScentAir. adresinden alındı
- SCENTGOURMAND (Yöneten). (2022). *Criteria for Rating Fragrance - Factors to Consider when Deciding your Purchase! [Sinema Filmi]*. <https://www.youtube.com/watch?v=IVmFmMRmrXk> adresinden alındı
- scentspiracy. (tarih yok). *Artificial Intelligence: Can AI Replace Professional Artists?* scentspiracy: <https://www.scentspiracy.com/blog/artificial-intelligence-can-ai-replace-professional-artists> adresinden alındı
- Sözlüğü, E. B. (tarih yok). *Önceden Eğitilmiş Model*. Encord: <https://encord.com/glossary/pre-trained-model-definition/> adresinden alındı
- Sözlüğü, E. B. (tarih yok). *Önceden Eğitilmiş Model*. Önceden Eğitilmiş Model: <https://encord.com/glossary/pre-trained-model-definition/> adresinden alındı
- Spina, L. (2025, Şubat 12). *Are social media and AI undermining the exclusivity of niche perfumery?* Istituto Marangoni: <https://www.istitutomarangoni.com/en/maze35/industry/are-social-media-and-ai-undermining-the-exclusivity-of-niche-perfumery> adresinden alındı
- Straits Research. (2024). *Perfume Market Size, Share & Trends Analysis Report By Product Type (Mass, Premium), By End-User (Men, Women, Unisex), By Distribution Channel (Online, Offline) and By Region(North America, Europe,*

APAC, Middle East and Africa, LATAM) Forecasts, 2025-2033. Straits Research.

The olfactive pyramid. (tarih yok). Parfums Sylvaine Delacourte: <https://www.sylvaine-delacourte.com/en/guide/the-olfactive-pyramid> adresinden alındı

Vinicius V. Santana, M. A. (2020, Mart 20). Optimal fragrances formulation using a deep learning neural network. *Computers and Chemical Engineering*.

Zion Market Research. (2022). *Perfume Market Size, Share, Analysis, Trends, Growth Report, 2030.* Amerika Birleşik Devletleri: Zion Market Research. 05 17, 2025 tarihinde alındı

EKLER**EK-1**

Kontrol Edilecek Hususlar	Evet	Hayır
Sayfa yapısı uygun mu?		
Şekil ve çizelge başlık ve içerikleri uygun mu?		
Denklem yazımları uygun mu?		
İç kapak, onay sayfası, Proje bildirim, özet, abstract, önsöz ve/veya teşekkür uygun yazıldı mı?		
Proje yazımı; Giriş, Kaynak Araştırması, Materyal ve Yöntem (veya Teorik Esaslar), Araştırma Bulguları ve Tartışma, Sonuçlar ve Öneriler sıralamasında mıdır?		
Kaynaklar soyadı sırasına göre verildi mi?		
Kaynaklarda verilen her bir yayına proje içerisinde atıfta bulunuldu mu?		
Kaynaklar açıklanan yazım kuralına uygun olarak yazıldı mı?		
Proje içerisinde kullanılan şekil ve çizelgelerde kullanılan ifadeler Türkçe'ye çevrilmiş mi? (Latince ve Özel kelimeler hariçtir)		
Projenin içindekiler kısmı, proje içerisinde verilen başlıklara uygun hazırlanmış mı?		

Yukarıdaki verilen cevapların doğruluğunu kabul ediyorum.

Unvanı Adı SOYADI

İmza

Öğrenci : Ömerhan SEZGİN

Danışman : Dr. Öğr. Üyesi ALİ ÇETİNKAYA