T.C. İSTANBUL KÜLTÜR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE GERÇEK OLMAYAN TÜKETİCİ YORUMLARININ TESPİTİ

YÜKSEK LİSANS TEZİ

Suat GÜLDAN

0809101045

Tezin Enstitüye Verildiği Tarih : 09 Temmuz 2014

Tezin Savunulduğu Tarih : 21 Temmuz 2014

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

Tez Danışmanı : Doç. Dr. Çağatay ÇATAL

Jüri Üyeleri : Doç. Dr. Banu DİRİ (Y.T.Ü)

Yrd. Doç. Dr. Akhan AKBULUT

TEMMUZ 2014

University : **Istanbul Kültür University**
Institute : **Institute of Sciences**
Department : **Computer Engineering**
Programme : **Computer Engineering**
Supervisor : **Assoc. Prof. Dr. Çağatay ÇATAL**
Degree Awarded and Date: **MSc – July 2014**

ABSTRACT

DETECTING DECEPTIVE CUSTOMER REVIEWS USING MACHINE LEARNING METHODS

Suat GÜLDAN - 2014

The competition among companies has been considerably increased in the recent years due to the significant developments in online shopping of services and the widespread usage of e-commerce. The product reviews became a primary factor shaping the buyers' decisions. Due to this factor, product reviews created a marketing area for fake reviews about products and services. In this thesis, a model which uses a multiple classifier system has been proposed to identify the negative deceptive customer reviews and the validation has been performed on a dataset which consists of hotel reviews. The proposed model has a better performance than the best model reported in literature for this problem. In this model, five classifiers have been applied by using majority voting combination rule. These classifiers are libLinear, libSVM, Sequential Minimal Optimization (SMO), Random Forest and J48. LibSVM and libLinear are two different implementations of support vector machines.

Keywords: Deceptive Review Detection, Support Vector Machines, Decision Trees, Multiple Classifier Systems, Vote, Machine Learning

Üniversite : **İstanbul Kültür Üniversitesi**
Enstitüsü : **Fen Bilimleri Enstitüsü**
Dalı : **Bilgisayar Mühendisliği**
Programı : **Bilgisayar Mühendisliği**
Danışmanı : **Doç. Dr. Çağatay ÇATAL**
Tez Türü ve Tarihi : **Yüksek Lisans – Temmuz 2014**

ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE GERÇEK OLMAYAN TÜKETİCİ YORUMLARININ TESPİTİ

Suat GÜLDAN – 2014

Çevrimiçi hizmetlerin gelişmesi ve e-ticaretin yaygınlaşması sonucu, firmalar arası rekabet giderek artmıştır. Bu bağlamda ürün yorumları, satın alıcıların kararlarını şekillendiren önemli bir faktör olarak ortaya çıkmıştır. Bu etkinin sonucu olarak ürün yorumları, ürün ve hizmetler hakkında aldatıcı yorumların yapılabileceği bir pazarlama alanı oluşturmuştur. Bu tezde, olumsuz aldatıcı tüketici yorumlarını tespit edebilmek üzere, çoklu sınıflayıcılı sistemler kullanılarak bir model önerilmiş ve önerilen modelin otel yorumları ile ilgili olarak hazırlanmış olan veri kümesinde geçerlenmesi sağlanmıştır. Önerilen model, üzerinde çalışılan problem için literatürdeki en iyi modelden daha yüksek performans sunmuştur. Bu modelde beş sınıflayıcı çoğunluk oylaması birleşim kuralına göre kullanılmıştır. Bu sınıflayıcılar libLinear, libSVM, ardışık minimal optimizasyon (SMO), Random Forest ve J48'dir. LibSVM ve libLinear, Destek Vektör Makinelerinin (DVM) iki farklı gerçeklemesi olarak bilinmektedir.

Anahtar Kelimeler: Aldatıcı Yorum Tespiti, Destek Vektör Makineleri, Karar Ağaçları, Çoklu Sınıflandırıcı Sistemler, Oylama, Makine Öğrenmesi

İÇİNDEKİLER

KISALTMALAR

Adaboost	: Adaptive Boosting
CF	: Confidence Factor
CV	: Cross Validation
DVM	: Destek Vektör Makineleri
GNU GPL	: GNU General Public License
NB	: Naive Bayes
SMO	: Sequential Minimal Optimization
MCS	: Multiple Classifier Systems
SVM	: Support Vector Machines

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

1. GİRİŞ

Çevrimiçinde tüketicinin hizmetine sunulan ürünler ve ürünler hakkındaki tüketici yorumları, diğer tüketicileri yönlendirmekte ve ürün hakkındaki düşüncesini etkilemektedir. Bu bağlamda ürünler hakkındaki diğer internet kullanıcılarının yorumları, kullanıcının ürünü satın alması veya almaması konusunda önemli ölçüde etkili olmaya başlamıştır [1, 2, 3]. İnternetin yaygınlaşması, bilgisayar ve mobil cihazlar ile ürünlere kolay ve hızlı ulaşımın olması, elektronik ticaretin büyümesine neden olmuş ve tüketici ürünlerindeki yorumlar da önemli bir rol üstlenmiştir. Hatta bir iş endüstrisi haline gelmiş, aldatıcı yorum hizmeti sunan şirketler oluşmuştur [4, 5, 6].

Aldatıcı yorumların tespitinde makine öğrenmesi metotları kullanmak başvurulan yöntemlerden birisidir. Bu tez çalışmasında makine öğrenmesi metotları kullanılmış olup, makine öğrenmesi ve sınıflandırma ile ilgili bilgiler takip eden bölümlerde sunulmuştur.

1.1. Makine Öğrenmesi

Farklı ortamlardan elde edilen verilerin bilgisayar ortamında farklı tekniklerle işlenerek sonuçlar çıkarılması durumuna makine öğrenmesi denilmektedir [7]. Makine öğrenmesinde birçok farklı algoritma kullanılabilmektedir. Bu algoritmaları sınıflandırma (classification), kümeleme (clustering) ve regresyon olarak üçe ayırabiliriz [7].

1.2. Doğal Dil İşleme ve Duygu Analizi

Doğal dil işleme; bilgisayar ortamında sayısal verilerin dilin yapısına göre anlamlandırılarak işlenmesine denir [8]. Doğal dil işleme alanında birçok çalışma yapılmaktadır. Her bir konuşma dilinin yapısı farklı olduğundan, doğal dil işleme uygulamaları dilin özelliklerine uygun yapılandırılmalıdır [8]. Duygu analizi de doğal dil işleme ile ilgili bir konu olup metin sınıflandırma ve duygu tespiti içerir. Duygu tespitinde konuşma dilindeki duyguları belirlemek için yapılan metin sınıflandırma çalışmaları bazı durumlarda zorlaşmaktadır. Bu çalışmada metin sınıflandırma bağlamında olumsuz aldatıcı yorumların tespit edilmesi için makine öğrenmesi yöntemlerine başvurulmaktadır.

Duygu kişinin içsel ve çevresel etkiler sonucunda ruh halinde meydana gelen sonuçtur. Ekman duygu üzerine yaptığı çalışmalar sonucunda temel duygu çeşitlerini tespit etmiştir. Bunlar öfke, iğrenme, korku, mutluluk, üzüntü ve sürpriz olmak üzere 6 tane temel duygu çeşidi vardır. İnsan yüzünde bu temel duygular anlaşılabilmektedir [9].

Parrot kitabında duygu çeşitlerini detaylı olarak incelemiştir. Duygunun sosyal psikoloji ile içi içe olduğunu tespit etmiştir. Yüzden fazla duygu çeşidini tanımlayarak, duygu çeşitleri için üç seviyeli bir ağaç yapısı oluşturmuş ve birbirine bağlantılı duyguları göstermiştir. [10]

Tablo 1-1: Parrot'un oluşturduğu duygu listesi [10]

Birincil Duygu	İkincil Duygu	Üçüncül Duygu
Aşk	eğilim	hayranlık, sevgi, aşk, sevgi, sevme, cazibe, ilgilenmek, hassasiyet, şefkat, duygusallık
	şehvet	uyarılma, arzu, şehvet, tutku, delicesine aşık
	özlem	özlem
Sevinç	neşe	eğlence, mutluluk, neşe, neşe, sevinç, neşe, neşe, sevinç, zevk, haz, sevinç, mutluluk, sevinme, sevinç, memnuniyet, ekstazi, coşku
	lezzet	coşku, heves, lezzet, heyecan, heyecan, canlılık
	memnuniyet	memnuniyet, zevk
	gurur	gurur, zafer
	iyimserlik	şevk, umut, iyimserlik
	büyülenme	büyülenme, tutsaklık
	rahatlama	rahatlama
Sürpriz	sürpriz	şaşkınlık, sürpriz, şaşkınlık irritasyon
	irritasyon	şiddetlenmesi, irritasyon, kışkırtma, sıkıntı, huysuzluk, somurtkanlık
Öfke	hiddet	hiddet, hüsran
	öfke	öfke, hiddet, gazap, düşmanlık, gaddarlık, kin, nefret, nefret, küçümseme, kin, intikam, sevmemek, kızgınlık
	iğrenme	iğrenme, tiksinti, aşağılama
	gıpta	gıpta, kıskançlık
	eziyet	eziyet
Üzüntü	acı	ızdırap, acı, incinen, keder
	üzüntü	depresyon, umutsuzluk, umutsuzluk, hüzün, hüzünlü, üzüntü, mutsuzluk, keder, üzüntü, keder, mutsuzluk, melankoli
	hayal kırıklığı	dehşet, hayal kırıklığı, hoşnutsuzluk
	ayıp	suçluluk, utanç, pişmanlık, vicdan azabı
	ihmal	yabancılaşma, izolasyon, ihmal, yalnızlık, reddedilme, gurbet, yenilgi, üzüntü, güvensizlik, utanç, aşağılanma, hakaret
	sempati	acıma, sempati
Korku	korku	alarm, şok, korku, korku, korku, terör, panik, histeri, çile
	sinirlilik	anksiyete, sinirlilik, gerginlik, huzursuz, endişe, sıkıntı, endişe, ürkmek

Bu duyguları belirtmek metinsel anlamda dilden dile farklılıklar göstermektedir. Bazı dillerde bir duyguyu ifade etmek için birden çok ifade biçimi varken bazı dillerde bir duyguyu ifade etmek için tek kelime kullanılabilir. Duygu kişinin kendini ifade etme tarzı ile doğrudan alakalı olduğu için; ses tonu, fiziksel tavrı ve konuşma biçimi aynı kelimeye farklı anlamlar yükleyebilmektedir. Bu gibi söyleyiş tarzı ile ilgili duygu çıkarımının sayısal metin sınıflandırmada yakalanması çok zordur. Bu durum metin sınıflandırmada bir kısıt olarak görülebilir.

Bing Liu "Sentiment Analysis and Opinion Mining" [11] adlı kitabında duygu analizini ayrıntılı olarak ele almaktadır. Kitabında duygu sınıflandırma ve görüş madenciliği hakkındaki tüm önemli fikirlere ve tekniklere yer vermeye çalışmıştır [11].

Nasukawa ve Ji duygu analizi konusunda kelimelerin anlamı ile bahsedilen nesne arasındaki ilişkilendirmeyi temel alan bir çalışma yapmışlardır. Bu çalışmada duygunun olumluluk ve olumsuzluk durumlarını %95 doğruluk oranında tespit edebilmişlerdir [12] .

Duygu tespiti literatürde en çok çalışma yapılan veri madenciliği alanlarından biri olmuştur. Buna karşın sahte yorum tespitinin nispeten zor olması ve bu konunun yeni bir araştırma alanı oluşturması nedeniyle bu konuda daha az çalışma mevcuttur.

1.3. Taslak

Tezin ilk bölümünde yapılan çalışmayla ilgili genel tanımlar ve özet bilgi sunulmaktadır. İkinci bölümde literatürde geçen ilişkili çalışmalar verilmektedir. Tezin üçüncü bölümünde tez çalışmasının altyapısını oluşturan makine öğrenmesi metotları ile ilgili bilgiler ortaya konulmaktadır. Bu bilgiler; makine öğrenmesinin genel tanımı, makine öğrenmesi algoritmaları, destek vektör makineleri ve uygulanan yöntemler olarak sıralanabilir. Dördüncü bölümde deneysel çalışmalarda kullanılan veri kümeleri ve özellikleri verilmektedir. Beşinci bölümde, deneysel çalışmalar ve bu çalışmaların neticesinde tespit edilen gözlemler. Altıncı bölümde sonuçlar ve gelecekte yapılabilecek olan çalışmalar verilmiştir. Son bölüm olan yedinci bölümde ise bu çalışmada kullanılan referanslar listelenmiştir.

2. LİTERATÜR TARAMASI

2.1. Metin Sınıflandırma ile İlgili Çalışmalar

Makine öğrenmesi metotları kullanılarak bir metnin bilinen sınıflar içinde hangisine daha yakın olduğunu tahmin etmeye metin sınıflandırma denir. Metin sınıflandırma üzerine birçok çalışma yapılmış olup, bazıları aşağıda verilmiştir:

Carbonell doktora tezinde politik konular üzerinde insan anlayışını doğal dil işlemede modellemeye çalışmıştır [13].

Berger ve arkadaşları doğal dil işleme alanında kelimelerin maksimum entropisini temel alan bir istatistiksel model sunmuştur [14].

Argamon-Engelson ve arkadaşları biçim tabanlı metin sınıflandırma çalışmalarını tanıtmışlardır. Bu çalışmalarında verilen metnin gazeteye veya magazine ait olması, promosyon amaçlı veya bilgi verici olması, anadili İngilizce olan bir kişi tarafından yazılmış olması gibi biçimleri tanımlamaya çalışmışlardır [15].

Turney kelime öbeklerinin anlamsal ortalama oryantasyonunu temel alan eğiticisiz bir algoritma öne sürmüş ve bu algoritmayı 4 farklı alana ait 410 yorum üzerinde test ederek %74 ortalama doğruluk elde etmiştir [16].

Pang ve arkadaşları, film yorumları üzerinde çalışmışlardır. Çalışmalarında film yorumlarını konu olarak değil, duygu olarak sınıflandırmak amacı ile 3 tane makine öğrenmesi (Naive Bayes, Maksimum Entropi Sınıflaması ve Destek Vektör Makinesi) algoritması kullanmış ve sonuçları karşılaştırmışlardır. Çalışmalarında duygu sınıflandırmanın konu sınıflandırmadan daha zor oluğunu ifade etmişler ve Turney'i destekleyici nitelikte duygunun metnin içindeki parçaların tamamı olmadığını raporlamışlardır [17].

Popescu ve Etzioni, "OPINE" adını verdikleri eğiticisiz bir model üzerinde çalışmışlardır. Ürün özellikleri, yorumlayıcı değerlendirmeleri ve ürünler arasındaki yakınlıktan bilgi çıkarımı yaparak modellerini oluşturmuşlardır. Yaptıkları testler sonucunda referans aldıkları diğer çalışmalardan daha yüksek kesinlik değerlerine ulaşmışlardır [18].

Hu ve Liu [19], çalışmalarında müşteri yorumlarında bahsedilen ürün ile ilgili özellikleri alarak veri madenciliği çalışması yapmışlardır. Kullandıkları özellik tabanlı özet çıkarımı

tekniği ile olumlu sonuçlara ulaşmışlardır. Yaptıkları testler sonucunda, beş ürün için cümle oryantasyonu tahmininde ortalama %84 oranında doğruluğa ulaşmışlardır [19].

Dave ve arkadaşları [20], olumlu ve olumsuz yorumları otomatik olarak tespit eden bir sınıflayıcı model üzerinde çalışmışlardır. Çalışmalarında ürünler için en uygun özellikleri ve metrikleri seçerek iyileşmiş sonuçlar elde etmişlerdir. Çalışmalarında veri kümesindeki değerlendirme tutarsızlıkları, belirsizlik ve karşılaştırma, dağınık veriler, çarpık dağılım gibi zorluklar tespit etmişlerdir [20].

2.2. Gerçek Olmayan Yorumların Tespiti ile İlişkili Çalışmalar

Jindal ve Liu, sahte tüketici yorumlarının tespiti ile ilgili ilk çalışmaları yapmışlardır. Çalışmalarında bir ürün ile ilgili veya farklı ürünlerde aynı yorumu yazan farklı kullanıcı hesaplarını tespit etmişler ve aldatıcı olup olmama durumlarına göre yorumları sınıflandırmışlardır. Kopya yorumların ve benzer yorumların tespitinde "Shingle" metodunu kullanmışlardır [21].

"Shingle" metodu ilk olarak Broder'ın çalışmasında kullanılmıştır. "Shingle" metodunda her bir doküman içinde birbirine yakın kelimeler gruplanır. Farklı belgeler için oluşturulan kelime grupları matematiksel bir formül ile karşılaştırılır. Sonuç olarak kelime grupları arasındaki benzerlik, iki dokümanın ne kadar benzer olduğunu 0-1 ölçeğinde göstermektedir [22] .

Jindal ve Liu sonraki makalelerinde [23], daha geniş bir araştırma gerçekleştirmişlerdir. Sahte yorumları genel olarak 3 tipe ayırmışlardır. Bunlar:

1- Gerçek olmayan yorumlar
2- Marka yorumları (markayı ön plana çıkarmak amaçlı yanıltıcı metinler)
3- Yorum olmayan fakat müşteri yanıltıcı tipteki metinler. (reklamlar, sorular, cevaplar vb. metinler).

Jindal ve Liu, bu 3 tip sahte yorumlardan ikinci ve üçüncünün kolaylıkla anlaşılabildiğini fakat birinci tip sahte yorumların tespit edilmesinin çok zor olduğunu belirtmişlerdir. Birinci tip sahte yorumların tespitinde önceki makalelerinde [21] de bahsettikleri Shingle metodunu kullanmışlardır. Veri seti olarak araştırmalarında Amazon.com web sitesinden aldıkları kullanıcı yorumlarını incelemişlerdir. Yorumların incelenmesinde kopya yorumları tespit ederken 3 farklı açıdan çalışma yapmışlardır:

- Farklı kullanıcıların aynı ürün hakkındaki benzer yorumları
- Aynı kullanıcının farklı ürünler hakkında benzer yorumları
- Farklı kullanıcıların farklı ürünler hakkındaki benzer yorumları

Tablo 2-1'de kopya yorumların tespitinde elde ettikleri sonuçlar görülmektedir. Jindal ve Liu sahte yorumların tespit edilmesi konusunda ilk incelemeleri yapmışlar fakat sahte yorumların tespitinin çok zor olduğunu, daha fazla çalışma yapılması gerektiğini belirtmişler ve tespit yöntemlerinin geliştirilmesi gerektiğini raporlamışlardır [23].

Tablo 2-1: Jindal ve Liu 'nun elde ettiği sonuçlar [23]

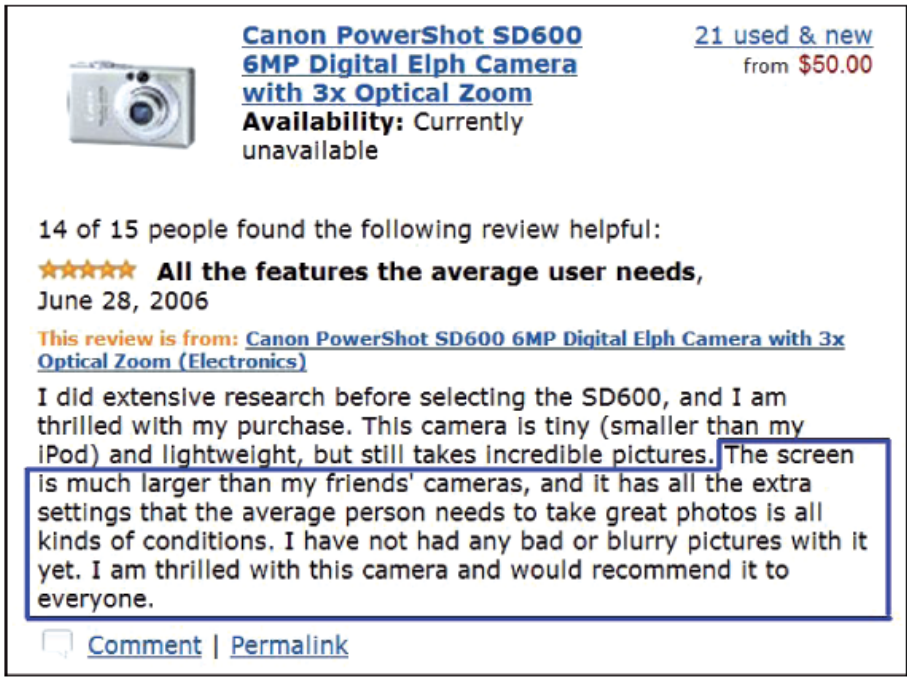
	Sahte Yorum Tipi	**Yorum Adetleri (Ürün Adetleri)**
1	Farklı kullanıcıların aynı ürün hakkındaki benzer yorumları	3067 (104)
2	Aynı kullanıcının farklı ürünler hakkındaki benzer yorumları	50869 (4270)
3	Farklı kullanıcıların farklı ürünler hakkındaki benzer yorumları	1383 (114)
	Toplam	55319 (4488)

Ott ve arkadaşları sahte yorumların tespiti ile ilgili ilk çalışmalarında "gold standard" [24] şeklinde tanımladıkları bir veri kümesi oluşturdular ve bu veri kümesi üzerinde çalışarak sahte yorumların tespitinde 5-katlı (5-fold) çapraz geçerleme yaparak, Naive Bayes (NB) ve Destek Vektör Makineleri (DVM) sınıflandırıcılarını incelemişlerdir [25].

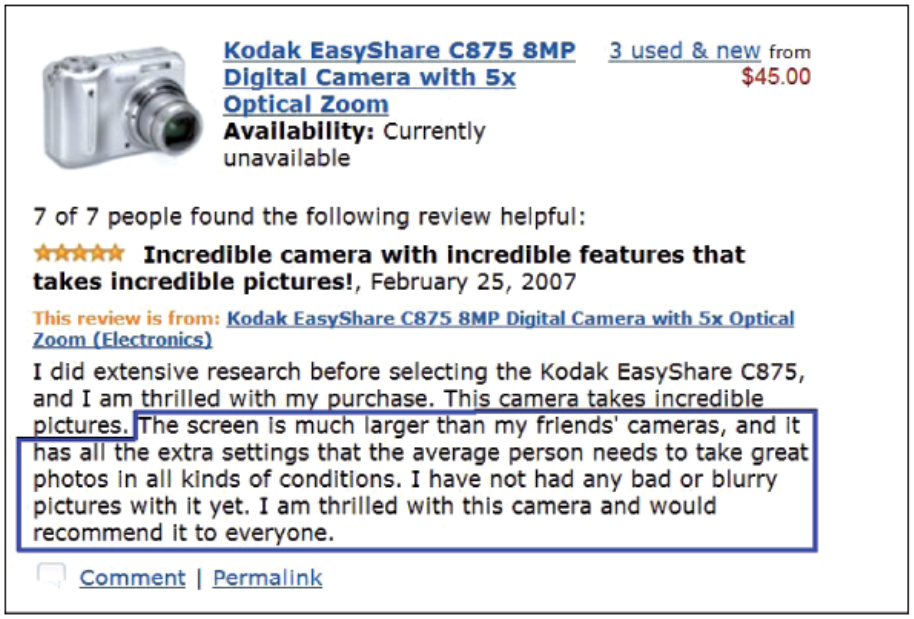
Ott ve arkadaşları diğer çalışmalarında, 6 popüler çevrimiçi yorum sitesinde (Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor ve Yelp) sahte yorum yaygınlığını araştırmışlardır. Araştırmaları sonucunda sahte yorumların gittikçe arttığını gözlemlemişlerdir. Sahte yorumların, yorum yazmanın kolay olduğu sitelerde daha yoğun olduğu bildirilmiştir [26].

Ott ve arkadaşları son çalışmalarında, ilk çalışmalarında [25] elde ettikleri veri kümesini [24] kullanarak olumsuz aldatıcı yorumların tespiti üzerinde çalışmışlardır. Aynı şekilde 5-katlı (5-fold) CV uygulayarak DVM yöntemini test etmişlerdir. Sonuç olarak %86 doğruluk oranıyla olumsuz sahte yorumların tespitine ulaşmışlardır [27].

Lau ve arkadaşları çalışmalarında hesaplamalı bir sahte yorum tespit modeli geliştirmişlerdir. Dil modellerinde olasılık fonksiyonu kullanmışlardır. “Sevmek” ve “hoşlanmak” kelimeleri gibi, kelimeler arasındaki ilişkileri göz önünde bulundurmuşlardır [28]. Aşağıdaki şekillerde (Şekil 2-1, Şekil 2-2, Şekil 2-3) Lau ve arkadaşlarının çalışmalarında verdikleri sahte yorum örnekleri görülmektedir.



Şekil 2-1: Sahte yorum-birinci örnek [28]



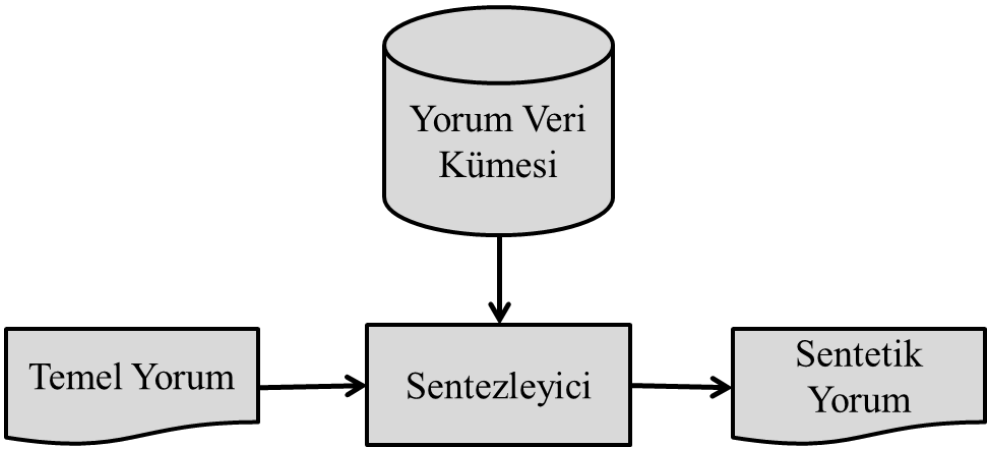
Şekil 2-2: Sahte yorum-ikinci örnek [28]

Şekil 2-3: Yakın anlamlı kelimeler için örnek [28]

Sharma ve Lin yaptıkları araştırmada beş teknik incelemişlerdir. İlk olarak, ürün derecelendirmelerini ve kullanıcı yorumlarını karşılaştırmışlardır. Derecelendirmeler ile yorumlar arasında bir tezat bulunursa bunu sahte yorum olarak ele almışlardır. İkinci olarak, yorumlardaki soru şeklindeki metinleri incelemişlerdir. Üçüncü olarak, yorumlarda bütün harfleri büyük harflerle yazılmış olan yorumları, belli bir durumu öne çıkarmak için bilinçli yazıldığından dolayı sahte olarak kabul etmişlerdir. Yorumlarda ayrıca karşılaştırmalı bir cümle kullanılmış ise bu durumu da sahte yorum olarak ele almışlardır. Son olarak yorumlarda belli bir web sayfasına bağlantı verilmişse sahte yorum olarak değerlendirmiştirler [29].

Chandy ve Gu [30], Apple uygulama mağazasındaki yorumlar üzerinde çalışmışlardır. Belirledikleri kurallara göre yorumları seçip etiketlemişlerdir. Doğrusal Gauss parametreleştirme kullanarak sahte yorumları tespit etmişler ve etiketlenmiş veri üzerinde karar verme ağaçlarından daha yüksek doğruluk değeri elde etmişlerdir [30].

Morales ve arkadaşları [31], ilk olarak gerçek yorumlar kullanarak otomatik sahte yorum üretip (Şekil 2-4'de model gösterilmiştir), yeni yorumların sahte yorum olma olasılıklarını tespit etmeye çalışmıştır. Fakat hata oranı yüksek çıktığı için bu model başarısız olmuştur. Ayrıca, New York şehrinde bulunan oteller için tripadvisor.com web sitesinden aldıkları kullanıcı yorumlarını, DVM ve Naive Bayes sınıflandırıcıları kullanarak sahte yorumları tespit etmeye çalışmışlardır. Sonuç olarak %22'lik bir hata oranını yakalayarak hali hazırda var olan %35-%48 aralığındaki hata oranını iyileştirmişlerdir [31].



Şekil 2-4: Otomatik sentetik yorum modeli [31]

Morales ve arkadaşları sonraki çalışmalarında modellerini geliştirmişler (Şekil 2-4) ve otomatik sentetik oluşturulan sahte yorumların tespitinde yeni bir sistem geliştirmişlerdir. Sahte yorumların tespitinde performansı %13 oranında iyileştirmişlerdir [32] .

Mukherjee ve arkadaşları [33], sahte yorumların tespitini farklı açıdan ele almışlar ve sahte yorumcuların davranışlarını incelemişlerdir. Modellerine "Author Spamicity Model" (ASM) ismini vermişlerdir. İlk olarak yazarın ve yorumun özelliklerini tespit etmişlerdir. Yazar özelliklerinde içerik benzerliği, yorum sayısı, yorum dağınıklığı ve ilk yorum derecesini ele almışlardır. Yorum özelliklerinde ise kullanıcıların kopya yorumlar, yüksek derece, derece sapması, erken zaman bölümü ve kötü derece durumlarını incelemişlerdir. Eğiticisiz Bayes çıkarım çatısı kullanarak sahte yorumları tespit etmeye çalışmışlardır. Sonuç olarak insan değerlendirmesinden daha etkili bir modele ulaşmışlardır [33].

Xie ve arkadaşları tekli yorumlar üzerinde bir çalışma gerçekleştirmiştir. Tekli yorumların bir ürünün satın alınmasında veya alınmamasında güçlü bir etkisi olduğunu gözlemlemişlerdir. Bir yorum şablonu ve çeşitli istatistikleri kullanarak çok boyutlu zaman serisi hazırlamışlardır. Zaman serileri üzerinde çalışabilen sahte yorum tespit algoritması tasarlamışlardır. Sonuç olarak tekli yorumlarda sahte yorumların tespiti için etkin sonuçlara ulaşmışlardır [34].

Bu tezde, çoklu sınıflayıcılı sistemleri (MCS) kullanarak sahte yorumların tespitini daha yüksek performansla gerçekleştirmeye çalıştık. Bu kapsamda öncelikle yöntemler problemler üzerinde tek tek denendi ve sonrasında farklı kombinasyonlar oluşturularak performansın iyileştirilebileceği oylama yönteminden yararlandık. Çoğunluk oylamasına göre farklı sınıflayıcıların bir arada yer aldığı farklı modeller, tez çalışması süresince incelendi ve WEKA üzerinde performans analizleri gerçekleştirildi. Yaptığımız literatür incelemesine göre, sahte yorumların tespitinde önerdiğimiz modeldeki gibi çoklu sınıflayıcılı sistemler literatürde önceden kullanılmamıştır. Bu yönüyle, özgün olarak 5 sınıflayıcıdan yararlanan bir model ilk kez ortaya konulmuş olup ulaşılan performans değeri Ott ve arkadaşlarının [27] modelinin performans değerinden daha yüksek olmuştur.

3. ALTYAPI

3.1. Makine Öğrenmesi

Bu tez çalışmasında makine öğrenmesi yöntemleri kullanılmıştır. Bu bakımdan makine öğrenmesi yöntemleri hakkında açıklayıcı bazı bilgiler aşağıda verilmiştir. Makine öğrenmesi algoritmaları bir matematiksel teoriye dayanarak geçmiş deneyimlerden elde edilmiş olan veriyi inceleyerek gelecek durumlar için kestirimler gerçekleştirir [7].

Makine öğrenmesi algoritmalarını kullanım türüne bağlı olarak eğiticili öğrenme (supervised learning), eğiticisiz öğrenme (unsupervised learning), yarı eğiticili öğrenme (semi-supervised learning), pekiştiricili öğrenme (reinforcement learning), uyum (transduction) ve öğrenmeyi öğrenme şeklinde altıya ayırabiliriz [35].

Bu tez çalışmasında; eğiticili öğrenme yöntemleri kullanılarak, girdi olarak verilen veri kümesinden eğitim süreci gerçekleştirildi ve test işlemleri çapraz geçerleme ile gerçekleştirildi. Tez çalışmasında kullanılan algoritmalar hakkında açıklayıcı bilgiler aşağıdaki başlıklar altında verilmiştir.

3.2. Naive Bayes

Naive Bayes sınıflandırıcısı temel istatistiksel sınıflandırıcılardan birisidir. İsmini ünlü matematikçi Thomas Bayes'den almıştır [36, 37]. Naive Bayes teoreminde her bir nitelik verilen sınıf içinde diğer niteliklerden bağımsız olarak kabul edilir [37]. Naive Bayes sınıflandırma; normal (Gauss), kernel, çok terimli ve çok değişkenli çok terimli dağılımlara sahiptir [38]. McCalllum ve Nigam, Naive Bayes dağılımlarının sonuçlarını karşılaştırdıkları çalışmalarında Naive Bayes çok terimli dağılımın daha iyi sonuçlar verdiğini tespit etmiştir [39]. Bayes teoreminin formülü aşağıda denklem 1'de verilmiştir.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{3.1}$$

Naive Bayes metin sınıflandırıcısının metin madenciliğinde nasıl kullanıldığını daha iyi anlamak için takip eden örneği inceleyelim.

Aşağıdaki gibi iki metnin veri kümesi olarak verildiğini varsayalım:

Metin 1: "Birçok bilgisayar mühendisliği bölümlerinde anlatılan temel yazılım dillerinden biri C++ dilidir."

Metin 2: "Birçok yazılım projesi C++ dili ile geliştirilmiştir."

Metin 3' de yukarıdaki metinlerden hangisine daha yakın olduğunu bulacağımız yeni bir metin olsun.

Metin 3: " Temel yazılım dili C++."

Naive Bayes sınıflandırıcısı için iki sınıf olduğunu varsayıyoruz. Herhangi bir kelimenin herhangi bir sınıfta bulunma olasılığı aşağıdaki gibi gösterilebilir:

$$P(ki|S) \tag{3.2}$$

Buradaki ifade şu anlama gelmektedir. S sınıfı gerçekleşen olaysa, ki kelimesinin bu olayda yer alma olasılığını gösterir. Bu durumda Naive Bayes sınıflandırma yönetimi kullanılırsa kelime bazlı sınıflandırıcı aşağıdaki denklem ile gösterilebilir.

$$P(X|S) = \prod_i (ki|S) \tag{3.3}$$

Şöyle ki; sistemde sınıflandırılmak istenen X metninin S sınıfında olma olasılığı, bu metindeki bütün kelimelerin S sınıfında olma olasılıklarının çarpımıdır. Metin 3'deki her bir kelimenin olasılıkları Metin 1 sınıfı ve Metin 2 sınıfı içindeki olasılıkları çarpılarak sonuç elde edilir. Çıkan sonuç hangi sınıfta daha büyük ise Metin 3 o sınıfa ait anlamına gelmektedir. Metin 1 ve Metin 2'de geçen kelimelerin listesi Tablo 3-1'de listelenmiştir. Bu listede görüldüğü gibi hangi kelimenin hangi sınıfta geçtiği bilgisi yer almaktadır. Liste kullanılarak kolaylıkla Metin 3'deki kelimelerin Metin 1 ve Metin 2'deki olasılıkları kolayca hesaplanabilir. Böylece elde edilen veri kümesinde Metin 1 ve Metin 2'den gelen toplam 15 kelime bulunuyor.

Tablo 3-1: Kelime listesi

Sıra	Terimler/Metinler	Metin 1	Metin 2
1	Birçok	1	1
2	bilgisayar	1	0
3	mühendisliği	1	0
4	bölümlerinde	1	0
5	anlatılan	1	0
6	temel	1	0
7	yazılım	1	1
8	dillerinden	1	0
9	biri	1	0
10	C++	1	1
11	dilidir	1	0
12	projesi	0	1
13	dili	0	1
14	ile	0	1
15	geliştirilmiştir	0	1

Bu listeye göre Metin 1'de 11 kelime bulunuyor ve hepsinin frekansı aynı. Bütün kelimeler için aşağıdaki şekilde hesaplanır.

$$P(ki|M1) = \frac{1}{11} \tag{3.4}$$

Aynı şekilde Metin 2 içinde bu durum geçerlidir ve 7 kelimenin tamamı için aşağıdaki şekilde hesaplanır.

$$P(ki|M2) = \frac{1}{7} \tag{3.5}$$

Neticede yeni gelen ve sınıflandırılmak istenen Metin 3 için bütün kelimelerin değerleri her bir sınıf için hesaplanır.

M3 ={ Temel, yazılım, dili, C++}

$$P(temel|M1) = \frac{1}{11}$$

$$P(yazılım|M1) = \frac{1}{11}$$

$$P(dili|M1) = 0$$

$$P(C++|M1) = \frac{1}{11}$$

Burada 0 çıkan değerleri düzeltmek için iki yöntemden birisi kullanılmaktadır. Ya normalleştirme yapılarak bu değer sistemden çıkarılır, ya da çok düşük bir değer konulur. Normalleştirme yaparsak aşağıdaki gibi bir sonuç elde edilir.

$$P(M1|M2) = \sum_{k=1}^{4} P\frac{(ki|M1)}{4} \tag{3.6}$$

$$\frac{\frac{1}{11} + \frac{1}{11} + 0 + \frac{1}{11}}{4} = \frac{3}{44} = 0{,}0681$$

İkinci metin için hesaplamalar aşağıdaki şekilde yapılır.

$$P(temel|M2) = 0$$

$$P(yazılım|M2) = \frac{1}{7}$$

$$P(dili|M2) = \frac{1}{7}$$

$$P(C++|M2) = \frac{1}{7}$$

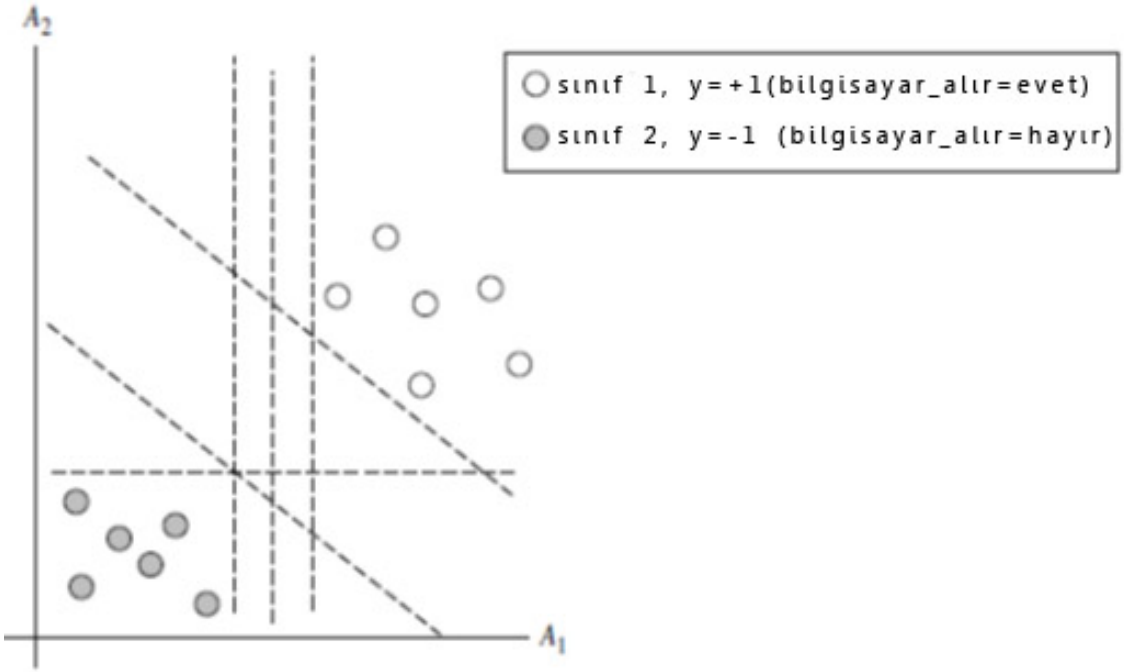
İkinci metin için normalleştirilmiş değer aşağıdaki gibi bulunur.

$$\frac{0 + \frac{1}{7} + \frac{1}{7} + \frac{1}{7}}{4} = \frac{3}{28} = 0{,}1071$$

Sonuç olarak 0,1071 değeri 0,0681 değerinden daha büyük olduğu için yeni gelen Metin 3, Metin 2'ye daha yakındır denilebilir. Metin 3 için bir sınıf seçilmesi gerekirse, Metin 2'nin bulunduğu sınıfta olduğu söylenebilir.

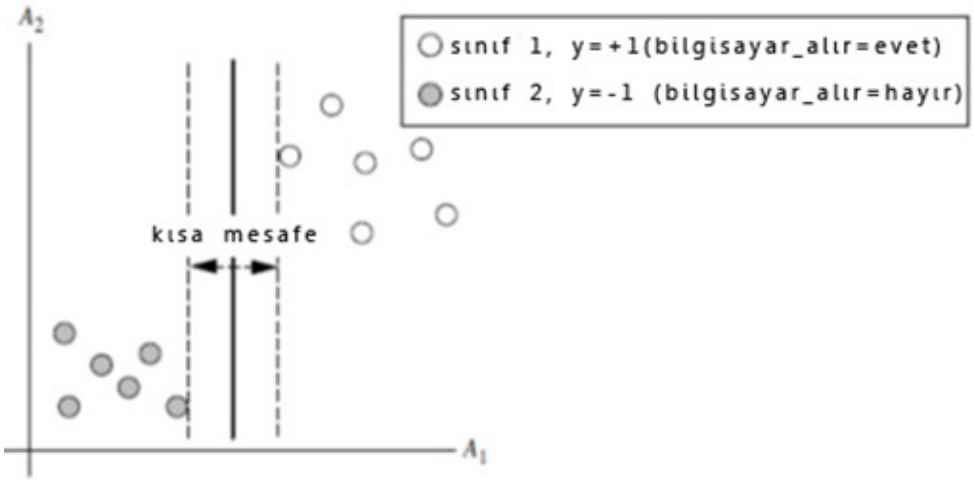
3.3. Destek Vektör Makineleri

Destek vektör makineleri eğiticili öğrenme yöntemlerindendir. Veriyi analiz etmede, model çıkarımında, sınıflandırma ve regresyon analizinde kullanılırlar. Dağınık bir veri üzerinde doğrusal en uygun hiperdüzlemi bulmaya çalışır. Bu hiperdüzlem iki grubun da üyelerini birbirinden ayıran en uzak mesafeyi içermelidir. Bu hiperdüzlem üzerindeki sınıflara ait noktalara destek vektörleri denir [40].

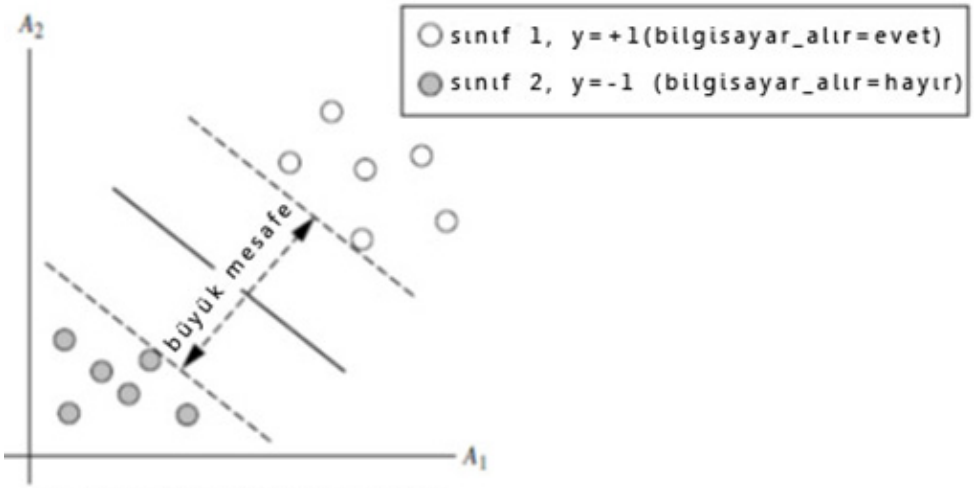


Şekil 3-1: DVM'de geniş olan hiperdüzlemin seçilmesi [40]

Şekil 3-1'deki gösterimde düzgün dağılmış veriler üzerinde iki sınıf arasında birden fazla hiperdüzlem bulunabildiği gösterilmiştir. Destek vektör makineleri küçük aralıklara sahip hiperdüzlemleri eleyip, en aşırı hiperdüzlemi seçer. Bunun sebebi ise hiperdüzlemin arasındaki uzaklık ne kadar büyük ise genel tutarlılığın da o derece doğru olmasıdır [40]. Şekil 3-2'de iki sınıf arasındaki kısa mesafeli düzlem ve uzun mesafeli düzlem gösterilmiştir. Şekil 3-3'de ise sonuç olarak seçilen en uzun mesafeli düzlem gösterilmiştir.

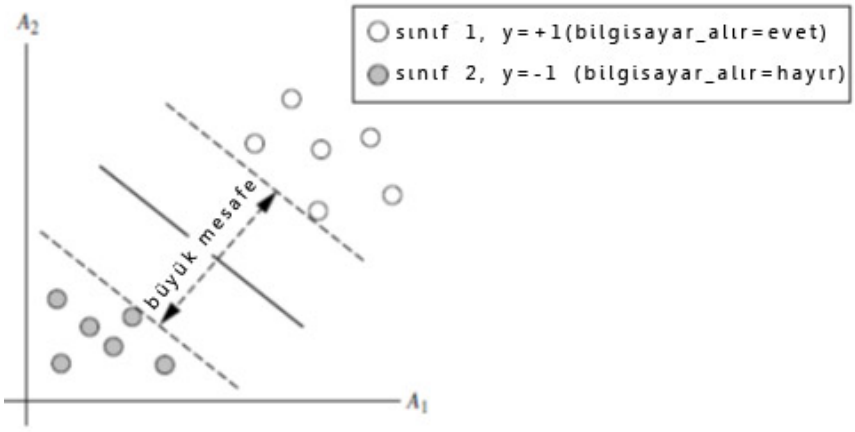


a)Kısa mesafeli hiperdüzlem



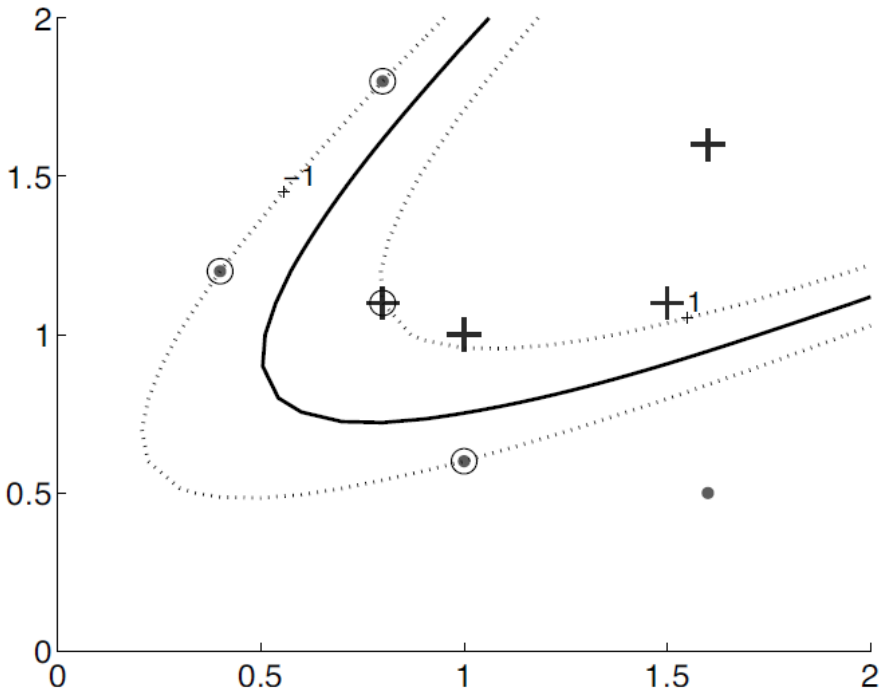
b)Büyük mesafeli hiperdüzlem

Şekil 3-2: Destek vektör makinelerinde hiperdüzlemlerin ayrı gösterimi [40]

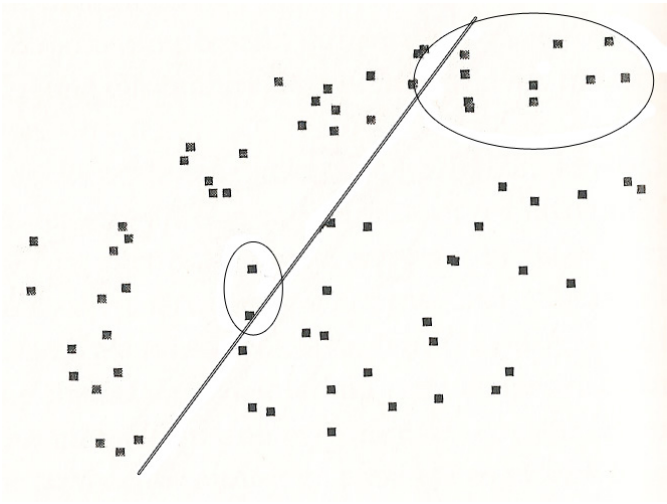


Şekil 3-3: Destek vektör makinelerinde seçilen büyük mesafeli düzlem [40]

Veri her zaman doğrusal dağınık şekilde bulunmamaktadır. Bu gibi durumlarda destek vektör makinelerinde veriler daha yüksek boyutlu bir uzaya taşınarak sınıflandırma işlemi yapılır. Yüksek boyutlu uzayda verilerin sınıflandırılması işlemi kernel fonksiyonları ile gerçekleştirilir [41]. Şekil 3-4'de doğrusal dağılmayan bir veri kümesi üzerinde sınıflandırma gösterilmiştir.

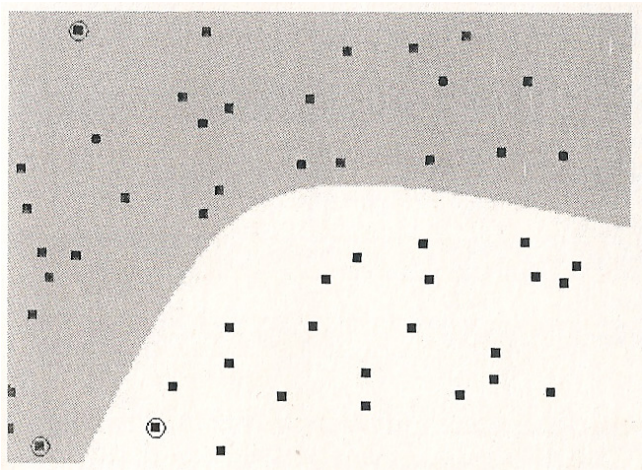


Şekil 3-4: DVM'de çok terimli kernel örneği [35].



Şekil 3-5: Doğrusal olarak ayrılmamış veri yapıları örneği [42]

Örneğin Şekil 3-5'daki gibi bir veri dağılımı söz konusu olduğunda doğrusal sınıflandırma yanlış sonuç verecekti. Doğru sınıflandırma yapabilmek için çekirdek fonksiyonları kullanarak Şekil 3-6'deki gibi doğru bir sınıflandırma yapılabilir [42].



Şekil 3-6: Doğrusal olmayan bir yöntemle sınıflandırma [42]

Bu çalışmada libLinear ve libSVM olmak üzere destek vektör makinelerinin WEKA veri madenciliği aracında iki farklı gerçeklemesi kullanılmıştır. LibLinear; L2-düzenlenmiş lojistik regresyonu, L2-kaybı ve L1-kaybı lineer destek vektör makinelerini destekler. LibLinear, libSVM 'deki birçok özelliği kullanır. Liblinear ve libSVM arasındaki göze çarpan farklardan biri libLinear gerçeklemesinin daha hızlı sonuç vermesidir [43].

3.4. Lojistik Regresyon

Lojistik regresyon istatistiksel bir sınıflandırma modelidir. Lojistik regresyon ikiterimli veya çokterimli olabilir. İki terimli lojistik regresyonda çıktı kümesi iki sınıf içerir. Çok terimli lojistik regresyonda ise çıktılar ikiden fazla sınıfa ait olabilir [44].

3.5. Ardışık Minimal Optimizasyon (SMO)

Ardışık minimal optimizasyon, optimizasyon problemlerinin çözümünde kullanılan John Platt tarafından 1998'de ortaya konulmuş bir algoritmadır. DVM'nin eğitiminde kullanılır. SMO çok büyük boyutlardaki veri kümeleri üzerinde ekstra hafızaya gerek duymadan çok hızlı analiz yapma imkânı sağlar [45]. SMO'da üç ana bileşen vardır.

Bunlar [45] :

- Uç değerleri çözmede kullanılan Lagrange çarpımını çözen analitik metot
- Çarpanları optimize etmek için sezgi
- Eşik değeri hesaplama için metot

3.6. Adaboost

Adaboost (Adaptive Boosting) makine öğrenmesinde kullanılan meta bir algoritmadır. Meta-algoritma öğrenmeyi öğrenen algoritma türlerine denilir [46]. Adaboost algoritması Yoav Freund ve Robert Schapire tarafından formüle edilmiştir. Adaboost algoritması güçsüz makine öğrenmesi algoritmalarını toplayarak güçlü bir algoritma ortaya çıkarmayı amaçlar [47]. Yoav Freund ve Robert Schapire adaboost meta-algoritması ile 2003'de Gödel Ödülü'nü kazanmışlardır [48].

AdaBoost zayıf sınıflayıcılardan güçlü bir sınıflayıcı oluşturan algoritmadır. Adaboost algoritmasının formülü aşağıda verilmiştir. Adaboost, zayıf ht(x) sınıflayıcılarını kullanarak güçlü bir sınıflayıcı oluşturmayı amaçlar. Formüldeki αt üssel hata fonksiyonunu minimuma indirgeyen bir değerdir ve ayrıca hesaplanır [49].

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t\,(x) \tag{3.6}$$

3.7. Bagging

Bagging metodu adabooost gibi meta bir algoritmadır. İngilizce "Bootstrap aggregating" kısaltılarak bagging adı verilmiştir. Eğer veri kümesi içinde gürültü oluşturan veriler varsa, bagging algoritması daha iyi sonuç verebilir. Bunun sebebi Bagging algoritmasının gürültülü (noisy) verileri dikkate almaması veya en aza indirgemesidir [50].

3.8. CVParameterSelection

CVParameterSelection algoritması adaboost ve bagging gibi meta bir algoritmadır. Bir diğer algoritmanın maliyet (cost) değerini optimize etmede kullanılır. İçi içe kullanılan algoritmalar için uygun değildir. Sadece tek bir sınıflayıcıda uygun sonuç verir [51].

3.9. Karar Ağaçları (J48 ve RandomForest)

Karar ağacı (decision tree learning) makine öğrenmesi yöntemlerinden birisidir. Literatürde karar ağacı öğrenmesinin sınıflandırma veya regresyon ağacı (regression tree) gibi uygulamaları vardır [42].

Karar ağaçlarında; veri kümesi özelliklere göre alt ağaçlara bölünür. Bu şekilde alt ağaçlarda da aynı işlem yapılarak tek bir sınıfa ait örnekler kalana dek bu işleme devam edilir ve yapraklara sınıf etiketi verilir [52].

Karar ağacı öğrenmesinde, eğitim kümesi çeşitli özelliklere göre özyinemeli (recursive) olarak bir alt küme kalmayana kadar alt kümelere bölünür. Yapılan bu işleme özyinemeli parçalama (recursive partitioning) ismi verilir [42].

Karar ağacı algoritmaları aşağıdaki şekilde özetlenebilir [42]:

- Rastgele Orman (Random Forest) : Birden fazla karar ağacı kullanarak sınıflandırma oranı yükseltilmeye çalışılır.
- Hızlandırılmış Ağaçlar: Hem sınıflandırma hem de regresyon (regression) problemlerinde kullanılabilir.
- Döndürme ağacı: Rastgele ağaca benzer olarak birden fazla ağaç kullanılır. Farklı olarak her bir ağaç öncelikle temel bileşen analizi (PCA) kullanılarak eğitilir. Bu eğitimde veri kümesinin rastgele seçilmiş bir alt kümesi kullanılır.
- ID3 algoritması
- C4.5 algoritması
- Chi-Kare Otomatik İlişki Tarayıcısı: Birden fazla seviyeye bölme işlemine izin veren bir sınıflandırma algoritmasıdır.
- Mars: Sayısal verilerin daha iyi işlenebilmesi için geliştirilmiş bir karar ağacı algoritmasıdır.

Karar ağacı algoritmasının avantajları [42]:

- Hızlı veri ön işlemesi gerektirir
- Hem sayısal hem sınıfsal verilerin işlenmesinde kullanılabilir
- Düşük hesaplama karmaşıklığına sahiptir.

Karar ağaçları avantajlarına rağmen bazı kısıtları vardır [42].

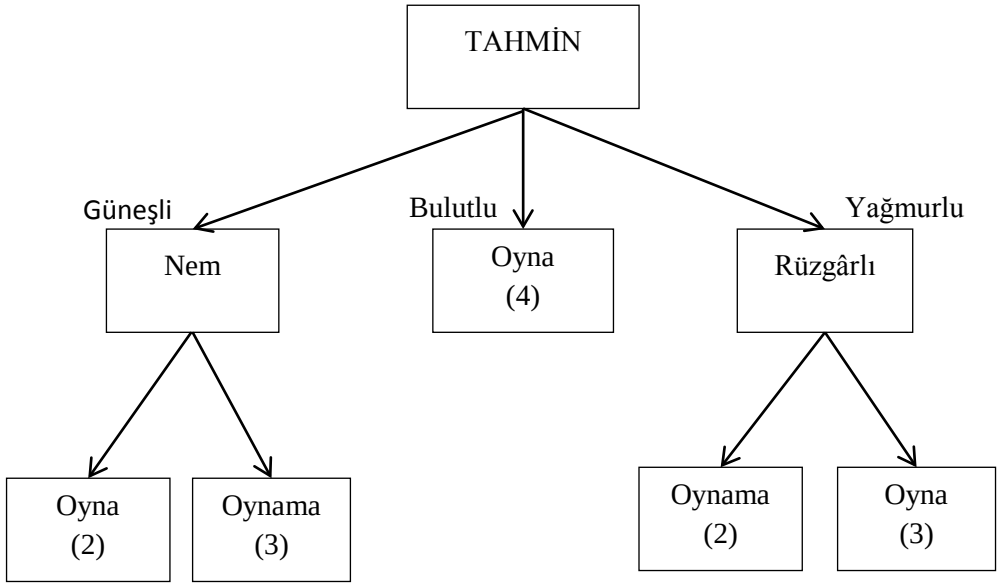
- İyileştirilmiş bir karar ağacı NP-tam karmaşıklığa sahiptir
- Ağaç oluşturulurken yapılacak bir hata, verinin özelliklerini modelleyemeyen bir ağaç yapısına neden olabilir

Bu tez çalışmasında karar ağaçlarından RandomForest ve J48 algoritmaları kullanıldı. J48, C4.5 algoritmasının Java dilinde uygulanmış haline verilen isimdir. Quinlan'ın, ID3 algoritmasını geliştirmesi sonucu C4.5 algoritması oluşmuştur. C4.5 eğitim veri kümesini kullanarak karar ağaçları oluşturur [53]. C4.5 algoritmasında iki aşama vardır. Birincisi karar ağacının oluşturulması ve ikincisi ise geçerli örnekler temel alınarak ağacın budanmasıdır. Ağaçtaki hataların azaltılması, alt ağaçların yapraklar ile değiştirilmesi şeklinde yapılır [54]. Tablo 3-2'de karar ağacı oluşturmada kullanılmak üzere örnek bir veri kümesi görülmektedir. Şekil 3-7' da bu veri setine ait karar ağacı görülmektedir [55].

Tablo 3-2: C4.5 Eğitim kümesinde kullanılacak veri kümesi [55].

Tahmin	**Sıcaklık**	**Nem**	**Rüzgârlı**	**Oyna (pozitif) / Oynama (negatif)**
Güneşli	85	85	Hayır	Oynama
Güneşli	80	90	Evet	Oynama
Bulutlu	83	78	Hayır	Oyna
Yağmurlu	70	96	Hayır	Oyna
Yağmurlu	68	80	Hayır	Oyna
Yağmurlu	65	70	Evet	Oynama
Bulutlu	64	65	Evet	Oyna
Güneşli	72	95	Hayır	Oynama
Güneşli	69	70	Hayır	Oyna
Yağmurlu	75	80	Hayır	Oyna
Güneşli	75	70	Evet	Oyna
Bulutlu	72	90	Evet	Oyna
Bulutlu	81	75	Hayır	Oyna
Yağmurlu	71	80	Evet	Oynama

Burada ağaç oluşturulurken tahmin seçenekleri ilk düğümler olarak seçilir ve şartlara bağlı olarak en son yaprağa ulaşana kadar devam edilir. Yaprağa ulaşıldığında, yapraktaki seçim yapılmış olur.



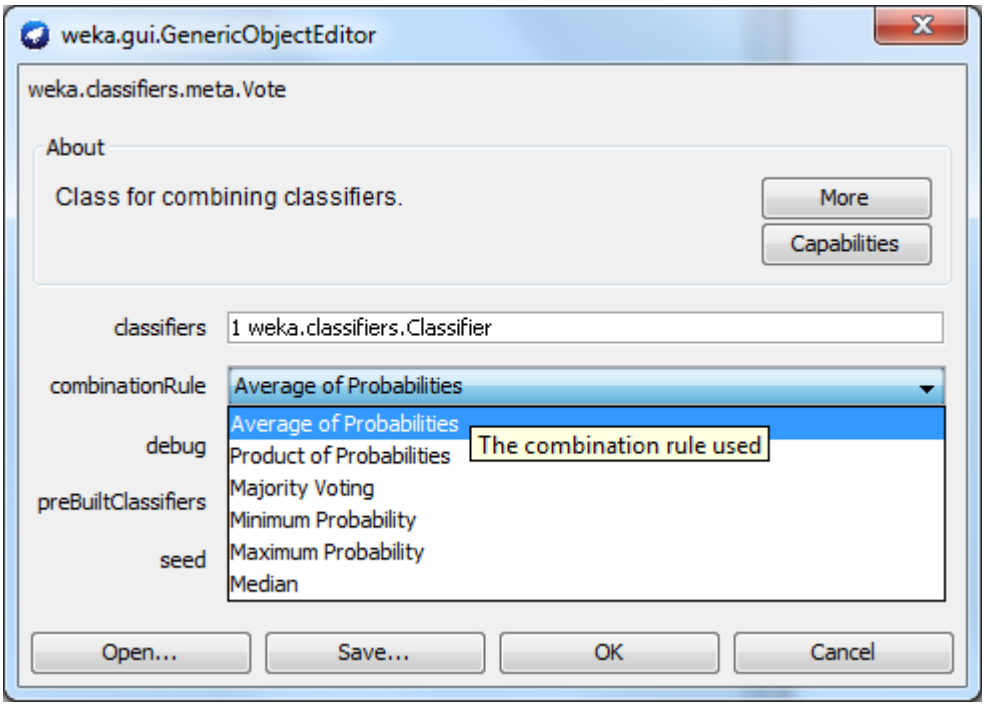
Şekil 3-7: C4.5 algoritması karar ağacı [55].

3.10. Oylama

Oylama (Vote) algoritması; CVParameterSelection algoritması, "adaboost" ve "bagging" gibi meta bir algoritmadır. Birden fazla algoritmayı oylama yaklaşımı ile kullanarak karar sürecini gerçekleştirir. Oylama algoritmasında, girdi olarak verilen algoritma çıktılarından sonuç çıkarmak için aşağıdaki seçenekler kullanılabilir:

- Olasılıkların ortalaması
- Olasılıkların çarpımı
- Çoğunluk oylaması
- Minimum olasılık
- Maksimum olasılık
- Medyan

Şekil 3-8'da WEKA programında Vote algoritmasının parametreleri gözükmektedir. Sonuçların nasıl belirleneceği "combinationRule" ayarından düzenlenebilmektedir.



Şekil 3-8: Vote algoritmasının parametreleri

Çalışmalarımızda, çoğunluk oylaması (majority voting) adı verilen birleşim kuralından yararlanıldı. Çoğunluğun kararına bağlı olarak modelimiz sınıf etiketini atamaktadır. Örneğin; yanıltıcı olan ve olmayan tüketici yorumları olmak üzere 2 sınıfı dikkate aldığımız zaman, 5 sınıflayıcının kararlarının çoğunluğuna göre model sonuç üretmektedir.

4. VERİ KÜMESİ

Bu tez çalışmasında; Ott ve arkadaşlarının [27] olumsuz sahte yorumları veri kümesi üzerinde ortaya koydukları modelin performansını iyileştirmek üzere farklı analizler gerçekleştirdik. Bu nedenle veri kümesi olarak Ott ve arkadaşlarının kullandıkları veri kümesini kullandık [27]. Veri setinin hazırlanması ile ilgili bilgi aşağıdaki başlıklar altında açıklanmıştır.

4.1. Veri Kümesinin Hazırlanması

Ott ve arkadaşları veri kümesini [25, 27] takip edilen şekilde hazırlamıştır. Tablo 4-1'de görüldüğü gibi veri kümesi toplamda 1600 adet yorum içermektedir. Veri kümesi, olumlu (pozitif) yorumlar ve olumsuz (negatif) yorumlar olmak üzere 2 klasörden oluşmaktadır ve her biri de kendi içinde 400 adet geçek yorum ve 400 adet sahte yorum olmak üzere toplam 800 adet yorum içermektedir. Pozitif yorum kümesi "gold standard" veri seti olarak kabul edilmiştir [25]. Fakat negatif veri kümesi yeterli sayıda yorum içinden seçilmediği için "gold standard" olarak kabul görmemiştir [26].

Tablo 4-1: Ott ve arkadaşlarının [25] veri kümesindeki yorum grupları ve adetleri

Pozitif Yorumlar (800)		**Negatif Yorumlar (800)**	
Gerçek Yorumlar (400)	Sahte Yorumlar (400)	Gerçek Yorumlar (400)	Sahte Yorumlar (400)

4.1.1. Pozitif Gerçek Yorumların Elde Edilmesi

Gerçek yorumlar dünya genelinde kullanılmakta olan TripAdvisor.com [56] sitesinden Şikago bölgesinde yer alan oteller hakkında kullanıcıların yazdığı yorumlar belli kısıtlara göre filtrelenerek alınmıştır [25]. Bu kısıtlar aşağıda açıklanmıştır:

- En çok yorum alan ilk 20 otele ait yorumlar alınmıştır.
- 1-5 ölçeğinde 5 yıldızlı olmayan yorumlar çıkarılmıştır.
- 150 kelimeden az olan yorumlar çıkarılmıştır. Bunun sebebi ise sahte yorum kümesinin en az 150 kelime içeriyor olmasıdır.
- İlk yorumunu yapmış olan kullanıcıların yorumları, sahte olma olasılığı yüksek olduğundan dolayı çıkarılmıştır [57].

4.1.2. Pozitif Sahte Yorumların Elde Edilmesi

Sahte yorumlar kalabalık bir çevrimiçi iş gücü servisi sağlayan AMT (Amazon Mechanical Turk) [58] web sitesinden yararlanılarak; toplam 400 sahte pozitif yorum oluşturulması için bir iş açılmıştır. Yorum yapacak olan AMT çalışanlarına, bazı kısıtlar ile yorum yapmaları zorlanmıştır [25].

Bu kısıtlar aşağıda listelenmiştir:

- Gerçek yorumların yapıldığı 20 otel için pozitif sahte yorum yapılması istenmiştir.
- 400 yorumun da her biri farklı bir çalışan tarafından yapılması sağlanmıştır.
- Sadece ABD'de yaşayan kullanıcıların yorum yapmaları sağlanmıştır. 30 dk içinde yorumlarını bitirmeleri istenmiştir.

4.1.3. Negatif Gerçek Yorumların Elde Edilmesi

Gerçek negatif (olumsuz) yorumlar popüler 6 web sitesinden (Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor ve Yelp), pozitif yorumların elde edildiği benzer kısıtlar kullanılarak elde edilmiştir [27].

Bu kısıtlar aşağıda açıklanmıştır [27]:

- En çok yorum alan ilk 20 otele ait yorumlar alınmıştır.
- 1-5 ölçeğinde 1 veya 2 yıldızlı yorumlar alınmıştır.
- Gerçek negatif yorumlarda kelime sayısı sahte yorumlardakine göre fazla olduğu için, yorumlardaki 150 kelimeden sonrası log-normal dağılıma göre çıkarılmıştır.
- İlk yorumunu yapmış olan kullanıcıların yorumları, sahte olma olasılığı yüksek olduğundan dolayı çıkarılmıştır [57].

4.1.4. Negatif Sahte Yorumların Elde Edilmesi

Negatif (olumsuz) sahte yorumlar ise pozitif sahte yorumlardaki gibi aynı şekilde kalabalık bir çevrimiçi iş gücü servisi sağlayan AMT (Amazon Mechanical Turk) [58] web sitesinden yararlanılarak, toplam 400 sahte negatif yorum yapılması için bir iş açılmıştır. Yorum yapacak olan AMT çalışanlarına bazı kısıtlar ile yorum yapmaları zorlanmıştır [27].

Bu kısıtlar aşağıda açıklanmıştır [27]:

- Gerçek yorumların yapıldığı 20 otel için negatif yorum yapılması istenmiştir.
- 400 yorumun da her biri farklı bir çalışan tarafından yapılması sağlanmıştır.
- ABD'de yaşayan kullanıcıların yorum yapmaları sağlanmıştır.
- 30 dk içinde yorumlarını bitirmeleri istenmiştir.

4.2. WEKA – Makine Öğrenmesi Aracı

Bu tezde veri kümesini işlemek ve sonuç çıkarmak için WEKA [59] yazılımı kullanılmıştır. WEKA makine öğrenmesi yöntemlerini içeren bir araç olarak geliştirilmiştir. Makine öğrenmesi aracı olmasına rağmen bazı veri madenciliği araçlarını da içermektedir. Birçok makine algoritmasını barındıran Java dilinde yazılmış faydalı bir veri madenciliği programıdır [59].

Waikato Üniversitesi tarafından geliştirilmiş olup, GNU GPL lisansı ile erişime sunulmuştur. Halen WEKA üzerinde yapılan geliştirmeler devam etmektedir. Yazılım geliştiricilere kendi yazılımlarını geliştirmeye veya WEKA kütüphanesini kullanmaya olanak sağlar. WEKA programı içindeki araçlar ile veri madenciliği kapsamında veriler üzerinde önişleme, kümeleme, sınıflandırma, ilişkisel kurallar, regresyon ve görselleştirme işlemleri yapılabilmektedir [59]:

Şekil 4-1: Weka programının ekran görüntüsü

Bu tez çalışmasında Şekil 4-1'de ana ekran görüntüsü bulunan Weka 3.7.10 sürümü kullanılmıştır. Weka programına, veri "arff" veya "csv" uzantılı dosyalar şeklinde girdi olarak verilebilmektedir. Ott ve arkadaşlarının [27] oluşturdukları "txt" uzantılı metin dosyaları "arff" uzantılı dosya formatına dönüştürülmüştür. Bu dönüşümü yapabilmek için ayrı bir program [60] kullandık. Bu program ve oluşturulan veri kümesi formatı bölüm 4.2.1'de açıklanmıştır.

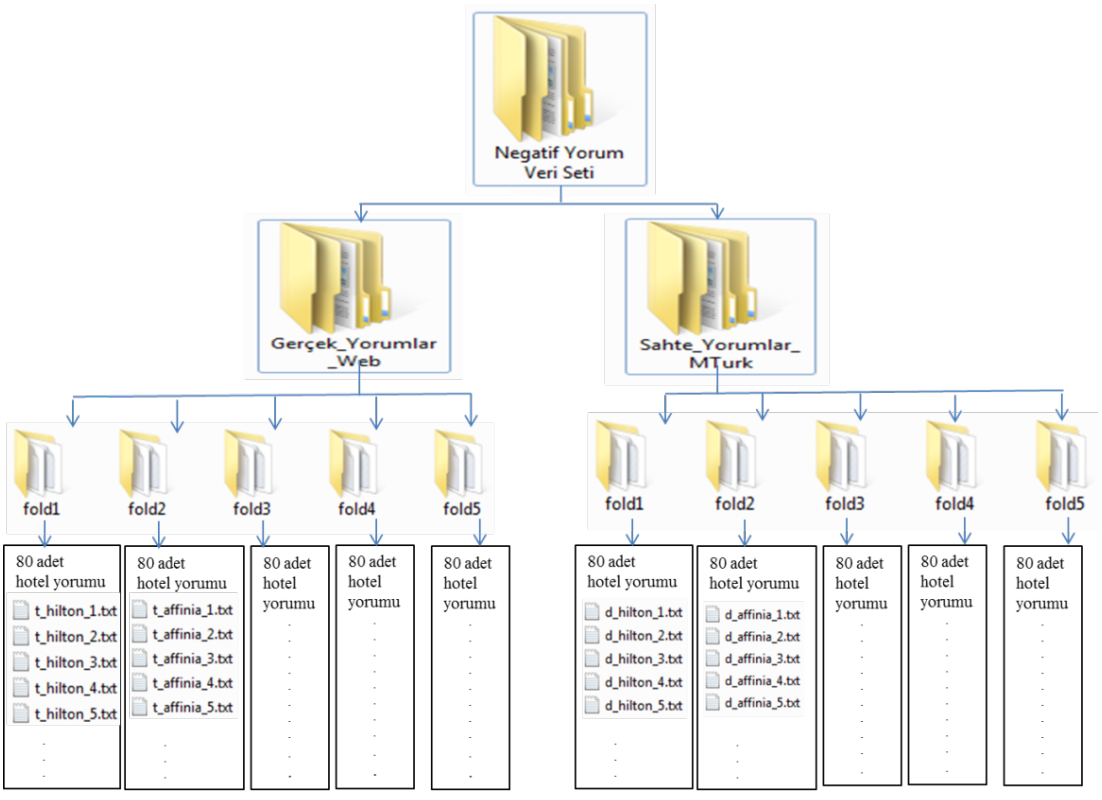
4.2.1. Veri Kümesi

Ott ve arkadaşlarının hazırlamış oldukları olumsuz (negatif) yorumlar veri kümesi [27], 400 gerçek ve 400 sahte olmak üzere 800 adet farklı "txt" uzantılı metin dosyalarında tutulmaktadır.

Tablo 4-2: Ott ve arkadaşlarının [27] hazırladıkları veri kümesine ilişkin bilgiler

Olumsuz (Negatif) Yorumlar	
Gerçek Yorumlar (400 adet "txt" uzantılı dosya)	Sahte (Aldatıcı) Yorumlar (400 adet "txt" uzantılı dosya)

Tablo 4-2'de veri kümesine ilişkin bilgiler görülmektedir. Sahte ve gerçek yorumlar olmak üzere iki sınıfımız olduğundan bu iki sınıf için iki ayrı klasör bulunmaktadır. Her bir klasörde ilgili yorumları içeren "txt" uzantılı dosyalar bulunmaktadır. Şekil 4-2'de negatif yorumlar veri kümesinin klasör yapısı görülmektedir.

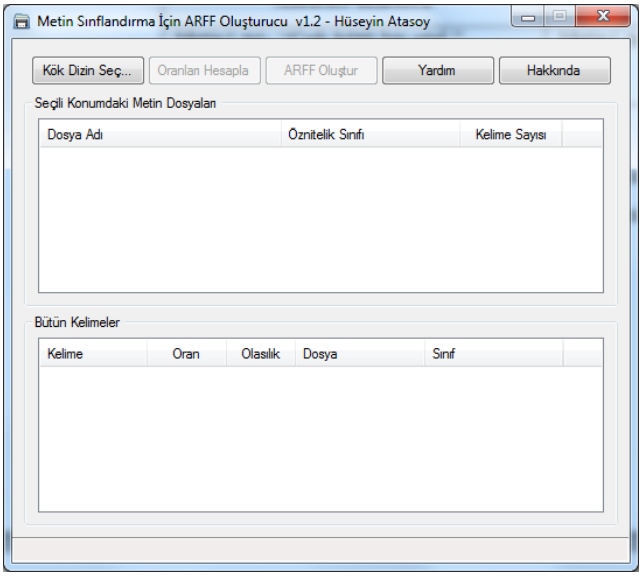


Şekil 4-2: Negatif yorumlar içeren veri kümesinin klasör yapısı

4.2.2. Veri Kümesinin Uygun Formatta Hazırlanması

Ott ve arkadaşlarının [27] hazırlamış oldukları metin tabanlı veri kümesi, bu hali ile WEKA programında kullanılamadığından, ilk olarak WEKA programına uygun hale

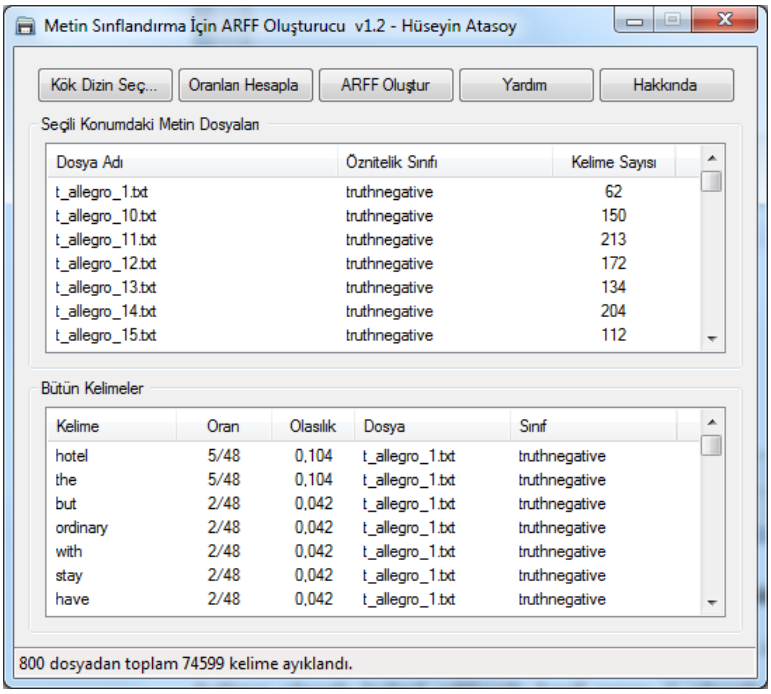
getirmek gerekiyordu. Bunun için “Metin Sınıflandırma için ARFF Oluşturucu” bir ara program kullanılmıştır. Metinleri WEKA programında sınıflandırabilmek üzere, WEKA programının gerektirdiği “arff” uzantılı dosyayı oluşturmak için kullanılan ara programın [60] ekran görüntüsü Şekil 4-3’de görülmektedir.



Şekil 4-3: “txt” dosyalardan “arff” uzantılı dosya oluşturan program [60]

Bu programa verilerin bulunduğu ana klasör kök dizin olarak gösterilip, minimum kelime olarak kabul edilecek harf sayısı 3 olarak girilir. Böylece 3’den daha az harften oluşan “is”,”he”, “it” gibi kelimeler elenmiş olur. İngilizce dilinin özelliğinden dolayı iki sınıfta 3’den daha az harf içeren kelimeler kullanılmış olabileceğinden sonuçları etkilememesi için yapılmıştır.

Bu programı çalıştırılarak; “txt” uzantılı dosyaları tarayıp kelime frekanslarını ve kelimelerin bulunma olasılıklarını hesaplayıp sınıflarına göre (gerçek, sahte) listelemesi sağlanmıştır. Şekil 4-4’de program çalıştırılarak metinlerden elde edilen kelime frekans ve olasılık değerleri arayüzde görülmektedir. Örneğin “hotel” kelimesi “t_allegro.txt” metin dosyasında 5/48 frekansı ile 0.104 olarak hesaplanmış ve gerçek olumsuz (truthnegative) yorum sınıfında kaydedilmiştir. Bu şekilde metin dosyalarındaki her bir kelime hesaplanmıştır.



Şekil 4-4: Kelime sayıları ve olasılıklarının hesaplanması

Sonra kelimeler öznitelik olarak alınır ve kelimelerin bulundukları metin içindeki bulunma oranlarını ve metnin bulunduğu sınıf "NegatifReviewDataSet.arff" dosyasına kaydedilir. Burada her sınıftan alınacak öznitelik sayısı maksimum seçilerek, tüm özniteliklerin WEKA programına uygun bir yapı ile kaydedilmesi sağlanır.

Programın çalışma mantığını ve oluşan "arff" dosyası formatını daha iyi açıklamak için aşağıdaki örnek sunulmaktadır. Örnek olarak sadece iki adet "txt" uzantılı dosyalardan biri gerçek, diğeri sahte klasörü altında oluşturulup, içerdikleri metinler Tablo 4-3'de gösterilmiştir.

Tablo 4-3: Veri kümesi klasör ve dosya yapısı

Aldatıcı (deceptive) Klasörü	Gerçek (truth) Klasörü
sahte1.txt- "This is worst hotel I ever seen."	*gercek1.txt-* "The hotel was very noisy."

Bu program metinlerden kelimeleri öznitelik olarak almakta ve tüm veri içinde kelimenin bulunma olasılıklarını çıkararak her bir dosya için tüm özniteliklerin olasılıklarını "arff" uzantılı olarak kaydetmektedir. Tablo 4-4'de oluşan veri kümesi gösterilmektedir.

Tablo 4-4: WEKA için hazırlanan MakaleOrnek.arff dosya içeriği

```
@RELATION "Makale_Ornek"

@ATTRIBUTE very REAL
@ATTRIBUTE noisy REAL
@ATTRIBUTE was REAL
@ATTRIBUTE the REAL
@ATTRIBUTE hotel REAL
@ATTRIBUTE ever REAL
@ATTRIBUTE seen REAL
@ATTRIBUTE this REAL
@ATTRIBUTE worst REAL
@ATTRIBUTE siniflar {truth,deceptive}

@DATA
0.2,0.2,0.2,0.2,0.2,0,0,0,0,truth
0,0,0,0,0.2,0.2,0.2,0.2,0.2,deceptive
```

Tablo 4-4'deki veri formatını şöyle açıklayabiliriz. @ RELATION anahtar kelimesi veri kümesine verilen ismi belirtmektedir. @ ATTRIBUTE anahtar kelimesi ile başlayan satırlar ise veri kümesindeki öznitelikleri belirtmektedir.

Örneğin; "@ATTRIBUTE very REAL" satırında "very" kelimesi özniteliği ve veri tipi ise "REAL" olarak temsil edilmiştir. Real olması bir kelimenin veri kümesinde olasılık değeri tipinin sayısal olduğunu ifade etmektedir. Öznitelikler bu şekilde alt alta sıralanmaktadır.

En sondaki öznitelik, veri kümesini sınıflandırmamız için kullanacağımız sınıfları ifade etmektedir. @DATA anahtar kelimesinden sonraki her bir satır ise veri kümesinin oluşturulduğu "txt" uzantılı dosyalardaki her bir satırın veri kümesindeki olasılık değerlerini ve sınıfı göstermektedir. Veri kümesindeki her bir kelime sırayla satır içerisinde aranır. İlgili satır içinde varsa, satır içinde bulunma adedi satırdaki toplam kelime sayısına oranı alınarak, veride bulunma olasılığı hesaplanmaktadır. Satır içinde yoksa olasılık değeri 0 olarak kaydedilmektedir. Hesaplanan olasılık

değeri, dosyanın başında belirtilen öznitelik sırasına göre arada virgül kullanılarak tüm özniteliklerin olasılıklarını gösterilmektedir. Metinde bulunmayan kelimeler için olasılık değeri 0 olarak gösterilir.

Tablo 4-5: "arff" uzantılı dosyasının açıklanması

@RELATION "MakaleOrnek"	**Gerçek Klasörü**-gercek1.*txt*
	The hotel was very noisy.
@ATTRIBUTE very REAL →	0.2 (very),
@ATTRIBUTE noisy REAL →	0.2(noisy),
@ATTRIBUTE was REAL →	0.2(was),
@ATTRIBUTE the REAL →	0.2(the),
@ATTRIBUTE hotel REAL →	0.2(hotel),
@ATTRIBUTE ever REAL →	0(ever),
@ATTRIBUTE seen REAL →	0(seen),
@ATTRIBUTE this REAL →	0(this),
@ATTRIBUTE worst REAL →	0(worst),
@ATTRIBUTE siniflar {truth,deceptive} →	truth
@DATA	
0.2,0.2,0.2,0.2,0.2,0,0,0,0,truth	
0,0,0,0,0.2,0.2,0.2,0.2,0.2,deceptive	

Tablo 4-5'de "gercek1.txt" dosyasında ilk satırdaki kelimelerin veri kümesine göre olasılıklarının detayı gösterilmektedir.

Örneğin: ilk satırdaki "0.2,0.2,0.2,0.2,0.2,0,0,0,0,truth" ifadesi gerçek (truth) klasörü altındaki gercek1.txt dosyasındaki "The hotel was very noisy." metninin öznitelik sırasına göre olasılıklarını ifade eder. Tablo 4-5'de görüldüğü gibi "very" özniteliği Gerçek Klasörü-gercek1.txt dosyasında 0,2 olasılığı ile bulunuyor. Aynı şekilde diğer özniteliklerin olasılıkları da yazılıyor ve en sonunda bu metnin ait olduğu sınıf yazılıyor. "seen", "this", ve worst" kelimeleri ise "gercek1.txt" dosyasında hiç bulunmadığından, bulunma olasılıkları "0" olarak kaydedilmiştir.

5. TESTLER

Ott ve arkadaşları [27] olumsuz sahte yorumlar hakkında %86 oranında doğruluğa ulaşmışlardır. Tablo 5-1'de Ott ve arkadaşlarının ulaştıkları sonuç gösterilmektedir. Tabloda kullanılan kısaltmaların anlamları şu şekildedir: D: Doğruluk (Accuracy), P: Duyarlılık (Precision), R: Kesinlik (Recall, Sensivity), F: P ve R'nin Harmonik ortalaması (F-Measure). Bu tez çalışmasında Ott ve arkadaşlarının çalışmalarını geliştirerek daha iyi sonuçlara ulaşmak amacıyla çoklu sınıflayıcılı sistemlerden yararlanılmıştır.

Tablo 5-1: Ott ve arkadaşlarının [25] DVM kullanarak ulaştığı sonuç

		Gerçek			**Sahte**		
Test Verisi	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
Olumsuz Yorumlar (800 adet)	**86,0**	86,4	85,5	85,9	85,6	86,5	86,1

5.1. Test Ortamı

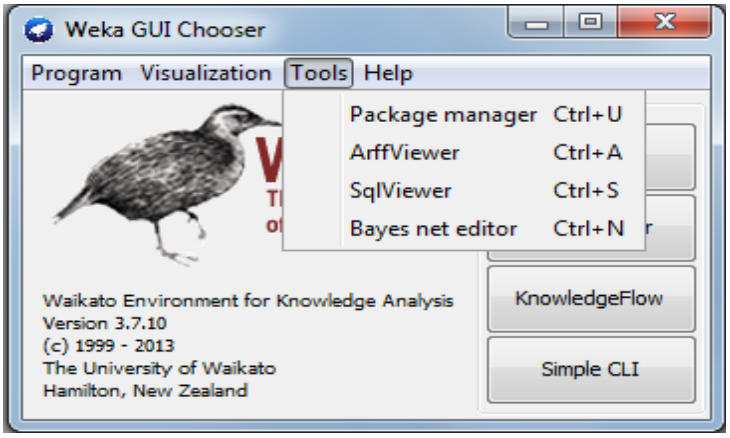
Yöntemleri test ve analiz etmek için Weka programı kullanılmıştır. Testler 2 farklı bilgisayarda gerçekleştirilmiş olup donanımsal özellikleri Tablo 5-2'de verilmiştir.

Tablo 5-2: Testlerin gerçekleştirildiği bilgisayarların donanımsal özellikleri

	1. bilgisayarın özellikleri	**2. bilgisayarın özellikleri**
İşlemci	Intel(R) Core(TM) i5-2540M CPU @ 2,60 Ghz	Intel(R) Core(TM) i5-2500 CPU @ 3,30 Ghz
Ram	4 GB	4 GB
Hard disk	232 GB 5200rpm	500 GB 7200rpm

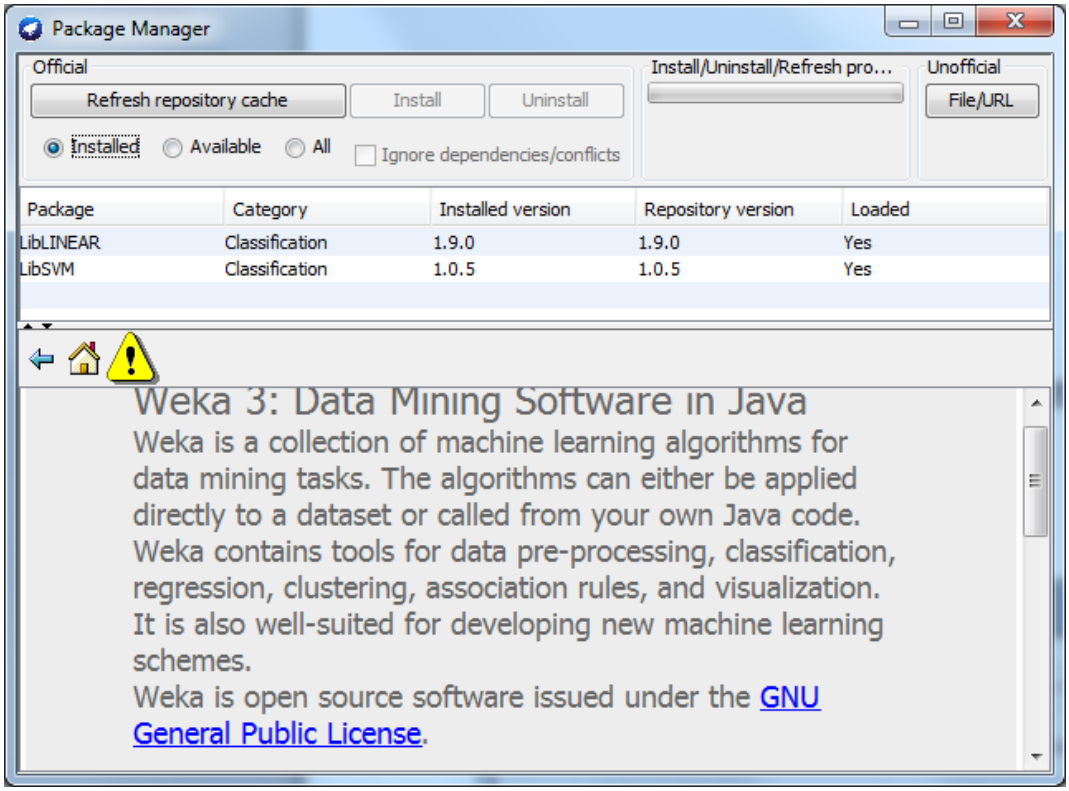
Weka programı GNU GPL lisansı ile dağıtıldığı için internet üzerinden ücretsiz olarak indirilebilir [61]. Testlerimizi yapabilmek için paket ekleme desteği sunan Weka 3.7.10 geliştirici sürümü indirilmiş ve bilgisayarlara kurulmuştur. İlk kurulumda gelmeyen libSVM ve libLinear paketleri, Weka programındaki paket yükleme yöneticisi arayüzü kullanılarak eklenmiştir. Şekil 5-1 ve Şekil 5-2'de; ek paketlerin yükleme yöneticisi ve yükleme ekranı gösterilmektedir.

Şekil 5-1'de WEKA arayüzünde bulunan "Tools" menüsünden paket yöneticisi menüsü seçilerek, indirilebilir WEKA paketlerinin listelendiği paket yöneticisi arayüzüne ulaşılabilmektedir.



Şekil 5-1: Weka programının paket yönetici menüsü

Şekil 5-2'de gösterildiği gibi paket yöneticisi ekranında çalışmamızda kullanacağımız "libLinear" ve "libSVM" paketlerini seçerek bilgisayara indirilmesi sağlanmıştır.



Şekil 5-2: libSVM ve libLinear paketlerinin yüklenmesi

5.2. WEKA ile Yapılan Testler

WEKA ile test yaparken veri kümesi olarak 4.2.1 numaralı başlıkta açıklanmış olan veri kümesi kullanılmıştır. Test sonuçlarının gösterildiği tablolarda kullanılan D,P,R ve F kısaltmalarının anlamları ve hesaplama formülleri (formüllerde kullanılan parametreler için karşıtlık matrisi Tablo 5-3'de gösterilmiştir.) aşağıdaki gibidir [62]:

Tablo 5-3: Karşıtlık Matrisi

	Öngörülen Sınıf	
Gerçek Sınıf	Gerçek Pozitif (TP)	Sahte Negatif (FN)
	Sahte Pozitif (FP)	Gerçek Negatif (TN)

D: Doğruluk (Accuracy)

$$\text{Doğruluk} = \frac{TP \ + TN}{TP + FP + FN + TN} \tag{5.1}$$

P: Kesinlik (Precision)

$$\text{Kesinlik} = \frac{TP}{TP + FP} \tag{5.2}$$

R: Hassaslık (Recall/Sensivity)

$$\text{Hassaslık} = \frac{TP}{TP + FN} \tag{5.3}$$

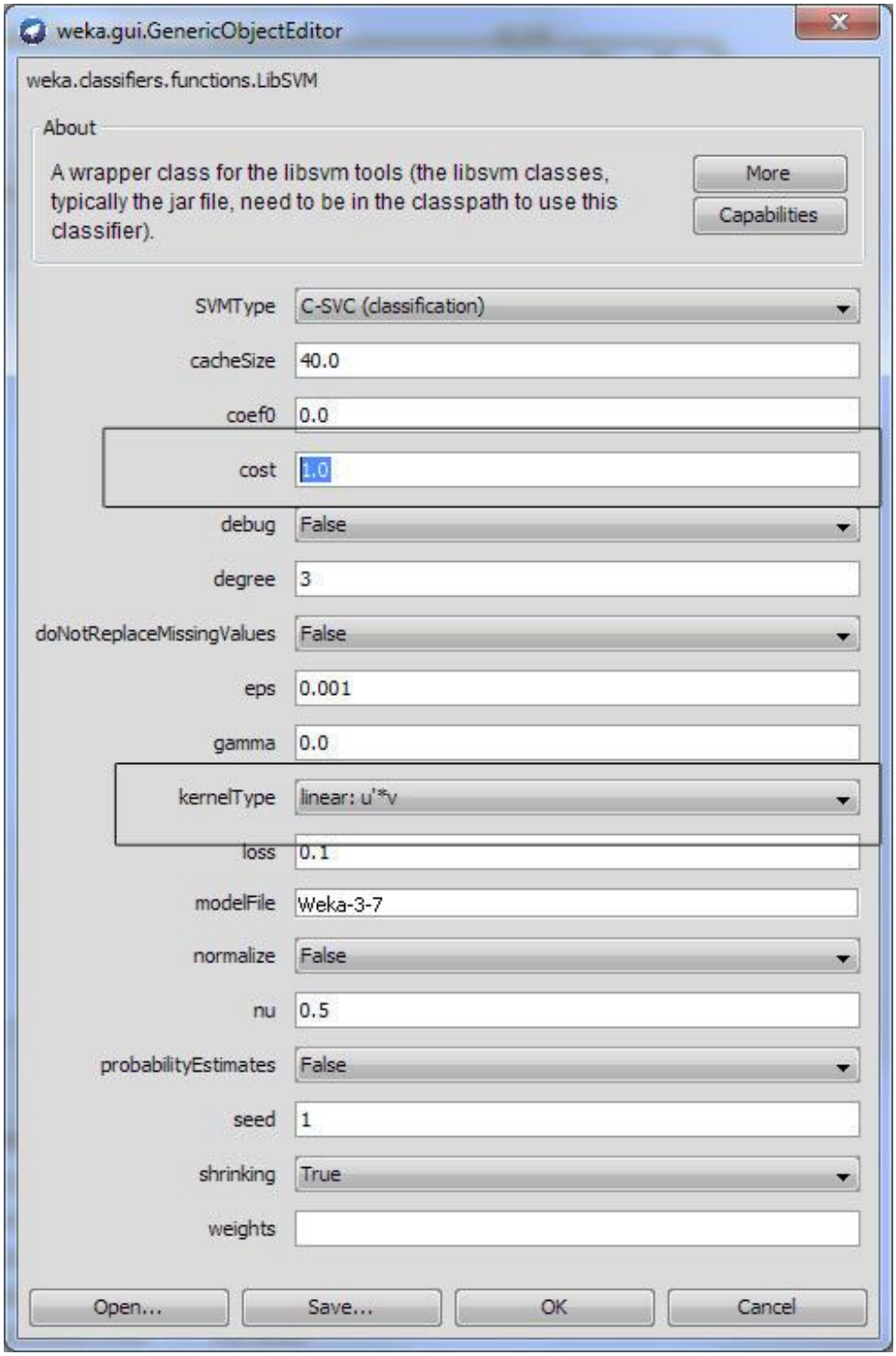
F: P ve R'nin Harmonik ortalaması (F-Measure)

$$\text{F} = 2 \cdot \frac{Kesinlik \ \cdot Hassaslık}{Kesinlik + Hassaslık} \tag{5.4}$$

Doğruluk, olumlu olarak tahmin edilen sonuçları ifade etmektedir. Kesinlik, doğruluk hesaplamasında tahmin edilen olumlu sonuçlar içinde hangilerinin kesin olumlu sınıf içinde olduklarını gösterir. Dolayısıyla kesinlik değeri doğruluk değerinden küçük veya eşit olabilir [62].

Veri kümesi ile ilk önce Ott ve arkadaşlarının ulaştığı %86 doğruluk oranı yeniden elde edilmeye çalışılmıştır (deney tekrarı). Ott ve arkadaşları 5-katlı çapraz geçerleme (5-fold cross validation) kullanarak DVM algoritmasında %86 doğruluğa ulaşmışlardır. Bu değere ulaşmak için WEKA 'da DVM algoritmasında testler

yapılmıştır. Şekil 5-3'de DVM algoritmasının parametre ekranı görülmektedir. Bu tez çalışmasında tüm testler boyunca DVM algoritmasında "kernelType" parametre değeri "linear:u'*v" olarak kullanılmıştır. DVM ile ilgili diğer testlerde "cost" parametresinin değeri değiştirilerek testler yapılmıştır. Ott ve arkadaşlarının çalışmasında parametrelerin tümü ayrıntılı olarak verilmediğinden deney tekrarı sürecinde farklı parametreler kullanılarak aynı performansa ulaşılmaya çalışılmıştır.



Şekil 5-3: Weka libSVM algoritmasının parametre seçim ekranı

5.2.1. Ott ve Arkadaşlarının [27] Ulaştığı Sonuç

DVM algoritmasında, "Cost" parametresini değiştirerek yapılan testler sonucunda %86 doğruluk oranına ulaşılmaya çalışılmıştır. %86 doğruluk oranına maliyet (cost) parametresi 60,5 yapıldığında ulaşılabilmiştir. Cost parametresi yayınlarında verilmediği için aynı sonucu üretmek oldukça güç olmuştur. Bu nedenle birebir aynı model kurulamadığından doğruluk, duyarlılık ve kesinlik değerleri farklı oluşmuştur.

Tablo 5-4: Ott ve arkadaşlarının [27] deneyinin tekrarlanması

		Gerçek			Sahte		
Maliyet	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
1,0	68,00	66,10	74,00	69,80	70,50	62,00	66,00
10,0	77,25	75,70	80,30	77,90	79,00	74,30	76,50
20,0	81,63	80,60	83,30	81,90	82,70	80,00	81,30
30,0	83,63	82,10	86,00	84,00	85,30	81,30	83,20
40,0	85,13	83,70	87,30	85,40	86,70	83,00	84,80
50,0	85,13	83,70	87,30	85,40	86,70	83,00	84,80
60,0	85,88	84,60	87,80	86,10	87,30	84,00	85,60
60,5	**86,00**	84,80	87,80	86,20	87,30	84,30	85,80

5.2.2. Naive Bayes Algoritması Testi

Olumsuz sahte yorumlar veri kümesi kullanılarak diğer makine öğrenmesi algoritmaları ile testler yapılmıştır. İlk önce Naive Bayes algoritması ile analiz edilmiştir. Naive Bayes algoritması analizinde NaiveBayesMultinomial kullanılmıştır [39]. Test sonucu Tablo 5-5'de de görüldüğü gibi %65,125 gibi düşük bir değerdir.

Tablo 5-5: NaiveBayesMultinomial test sonucu

	Gerçek			Sahte		
D %	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
65,125	95,50	31,80	47,70	59,10	98,50	73,90

5.2.3. DVM Algoritması Testi - Farklı Maliyet Değerleri

Sonraki testlerde DVM algoritmasının maliyet (cost) parametresine farklı değerler verilerek testler yapılmıştır. Daha önceki DVM testleri Ott ve arkadaşlarının 86% değerine ulaştıkları parametreleri tespit etmek amacıyla yapılmıştır. Buradaki testler performansı iyileştirmek için gerçekleştirilmiştir. Tablo 5-6'de görüldüğü gibi 75, 80 ve 90 maliyet değerlerinde en yüksek tutarlılık yakalanmış fakat modellerin P, R ve F değerlerinde küçük farklılıklar tespit edilmiştir.

Tablo 5-6: DVM Algoritmasında farklı maliyet değerleri ile test sonuçları

		Gerçek			Sahte		
Maliyet	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
70,0	87,00	85,70	88,80	87,20	88,30	85,30	86,80
75,0	**87,13**	85,80	**89,00**	**87,40**	**88,60**	85,30	86,90
80,0	**87,13**	86,00	88,80	87,30	88,40	85,50	86,90
85,0	87,00	86,10	88,30	87,20	87,90	85,80	86,80
90,0	**87,13**	**86,30**	88,30	87,30	88,00	**86,00**	**87,00**
95,0	86,88	85,90	88,30	87,10	87,90	85,50	86,70
100,0	86,63	85,60	88,00	86,80	87,70	85,30	86,40
105,0	87,00	85,70	88,80	87,20	88,30	85,30	86,80
110,0	87,00	85,70	88,80	87,20	88,30	85,30	86,80
120,0	86,50	85,30	88,30	86,70	87,80	84,80	86,30
130,0	86,25	85,00	88,00	86,50	87,60	84,50	86,00
140,0	85,88	84,90	87,30	86,10	86,90	84,50	85,70
150,0	86,00	84,80	87,80	86,20	87,30	84,30	85,80
160,0	86,13	85,00	87,80	86,30	87,30	84,50	85,90
170,0	86,13	85,00	87,80	86,30	87,30	84,50	85,90

5.2.4. SMO Algoritması Testi

Bir meta algoritma olan SMO algoritmasında çekirdek (kernel) değiştirilerek yapılan testlerin sonuçları bu bölümde gösterilmiştir. Çekirdek olarak "PolyKernel" seçimi daha yüksek performans vermiştir. Sadece gerçek yorumların tespitinde NormalizedPolyKernel çekirdek tipi daha iyi sonuç vermiştir fakat sahte yorumlar üzerinde çalıştığımız için "PolyKernel" çekirdeği daha iyi sonuçlar verdiğinden, sonraki oylama algoritması testlerinde "PolyKernel" çekirdeğini seçtik. Tablo 5-7'da SMO algoritması test sonuçları gösterilmektedir.

Tablo 5-7: SMO meta algoritması test sonuçları

		Gerçek			Sahte		
Çekirdek (Kernel)	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
PolyKernel	**84,25**	**84,40**	84,00	**84,20**	**84,10**	**84,50**	**84,30**
RBFKernel	77,00	75,20	80,50	77,80	79,00	73,50	76,20
NormalizedPolyKernel	79,13	75,70	**85,80**	80,40	83,60	72,50	77,60
PUK	63,75	62,70	67,80	65,10	64,90	59,80	62,20

5.2.5. Random Forest Algoritması Testi

Bir karar ağacı algoritması olan Random Forest algoritmasında ağaç ve çekirdek parametreleri değiştirilerek testler yapılmıştır. Tablo 5-8'de Random Forest algoritması test sonuçları gösterilmiştir. Testler sonucunda en yüksek değer 380 ağaç ve 60 çekirdek seçilerek gerçekleştirilen test olmuştur.

Tablo 5-8: Random Forest algoritması test sonuçları

		Gerçek			Sahte		
Parametreler	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
Trees 10, Seed 1	67,125	66,00	70,50	68,20	68,40	63,80	66,00
Trees 50, Seed 100	79,000	81,50	75,00	78,10	76,90	83,00	79,80
Trees 50, Seed 200	79,750	79,60	80,00	79,80	79,90	79,50	79,70
Trees 100, Seed 50	79,625	84,00	73,30	78,20	76,30	86,00	80,80
Trees 200, Seed 50	82,625	87,40	76,30	81,40	78,90	89,00	83,70
Trees 300, Seed 50	82,000	86,40	76,00	80,90	78,60	88,00	83,00
Trees 400, Seed 50	82,125	88,10	74,30	80,60	77,80	90,00	83,40
Trees 400, Seed 60	83,875	89,50	76,80	82,60	79,60	91,00	84,90
Trees 300, Seed 60	84,500	89,20	78,50	83,50	80,80	90,50	85,40
Trees 300, Seed 55	82,750	87,60	76,30	81,60	79,00	89,30	83,80
Trees 300, Seed 55	84,125	88,90	78,00	83,10	80,40	90,30	85,00
Trees 350, Seed 60	84,125	90,30	76,50	82,80	79,60	91,80	85,20
Trees 380, Seed 60	**84,750**	90,60	77,50	83,60	80,30	92,00	85,80

5.2.6. J48 Algoritması Testi

Karar ağacı algoritması olan J48 ile yapılan testler Tablo 5-9'de gösterilmiştir. Tablodaki CF kısaltması güven faktörü (confidence factor) parametresinin aldığı değerleri göstermektedir. J48 algoritmasında en iyi performans CF değeri 0,1 iken %68,5 doğruluk değeri ile elde edilmiştir.

Tablo 5-9: J48 algoritması test sonuçları

		Gerçek			Sahte		
CF	D %	P %	R %	F %	P %	R %	F %
0,1	**68,75**	67,80	71,50	69,60	69,80	66,00	67,90
0,2	68,63	67,60	71,50	69,50	69,80	65,80	67,70
0,25	68,25	67,40	70,80	69,00	69,20	65,80	67,40
0,5	68,13	67,40	70,30	68,80	68,90	66,00	67,40
1	68,13	67,40	70,30	68,80	68,90	66,00	67,40
2	68,13	67,40	70,30	68,80	68,90	66,00	67,40

5.2.7. Adaboost Algoritması Testi

Weka programında Adaboost algoritmasının gerçeklemesi olan "AdaboostM1" kullanılmıştır. Bu sınıflayıcı içinde diğer sınıflayıcılar kullanılarak yapılan testler sonucunda elde edilen en iyi değerler Tablo 5-10'da gösterilmiştir. Yapılan testler sonucunda en iyi performans %85,25 doğruluk oranıyla libLinear sınıflayıcısında maliyet değeri 120 seçilerek oluşturulan modelle elde edilmiştir.

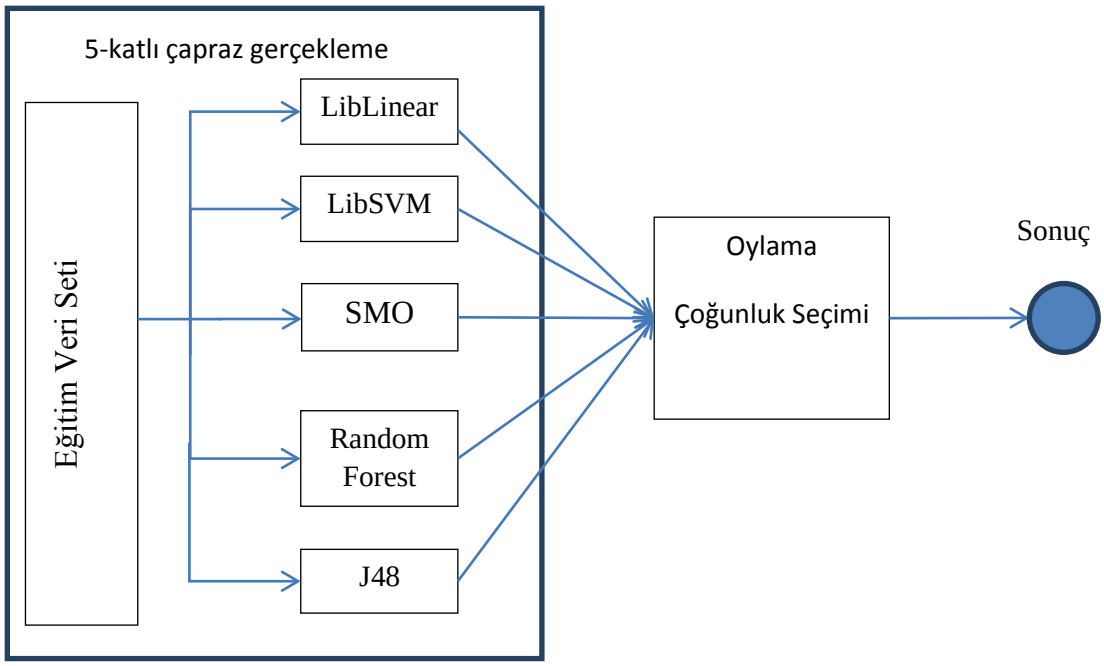
Tablo 5-10: Adaboost algoritması test sonuçları

		Gerçek			Sahte		
Sınıflayıcı	D %	P %	R %	F %	P %	R %	F %
LibLinear Cost:1	78,25	79,00	77,00	78,00	77,60	79,50	78,50
LibLinear Cost:120	**85,25**	84,70	85,80	85,20	85,60	84,50	85,00
libSVM Cost 60,5	84,50	84,80	84,00	84,40	84,20	85,00	84,60
libSVM Cost 120	84,00	84,70	83,00	83,80	83,30	85,00	84,20
BayesMultimonial	79,38	92,40	64,00	75,60	72,50	94,80	82,10
J48	72,88	72,80	73,00	72,90	72,90	72,80	72,80
RandomForest	73,38	71,30	78,30	74,60	75,90	68,50	72,00

5.2.8. Oylama Algoritması Testi

Oylama algoritmasında en iyi doğruluk değerlerini veren sınıflayıcıları kullanarak, çoğunluk oylamasına göre sonuç alınmıştır. Şekil 5-4'de beşli sınıflayıcı kullanarak oluşturduğumuz model gösterilmektedir. DVM, Naive Bayes, SMO, Random Forest, J48 ve Adaboost algoritmaları ile yapılan testlerde en yüksek değeri veren parametreler kullanılarak oylama algoritmasına girdi olarak verilmiştir. Sınıflayıcılar birbirleri arasında çeşitli

kombinasyonlarda oylama algoritmasında girdi olarak kullanılarak birçok test yapılmıştır. Ortaya konulan model Şekil 5-4'de resmedilmektedir.



Şekil 5-4: Oylama algoritmasının çalışma şekli

Çoğunluk oylamasında, sınıflayıcılar Tablo 5-11'da değerleri gösterilen en iyi sonuç veren parametreleri ile üçlü ve beşli kombinasyonlarla test edilmiştir. Tek bir sınıflandırma algoritması kullanılarak yapılan testlerin sonuçlarından daha yüksek sonuçlar elde edilmiştir. En yüksek doğruluk değeri %88,125 olup libLinear, libSVM, SMO, Random Forest ve J48 olmak üzere beşli sınıflayıcı içeren modelin testlerinde tespit edilmiştir.

Tablo 5-11: Sınıflayıcıların en iyi performans değerleri

			Gerçek			Sahte		
Sınıflayıcı	**Parametreler**	**D %**	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**
libLINEAR	Cost: 120	**87,125**	86,10	88,50	87,30	88,20	85,80	86,90
libSVM	Cost: 90,0	**87,125**	86,30	88,30	87,30	88,00	86,00	87,00
SMO	PolyKernel	**84,250**	84,40	84,00	84,20	84,10	84,50	84,30
Random Forest	Trees 380, Seed 60	**84,750**	90,60	77,50	83,60	80,30	92,00	85,80
J48	Confidence Factor :0,1	**68,750**	67,80	71,50	69,60	69,80	66,00	67,90
Vote (oylama)	**libLINEAR libSVM SMO Random Forest J48**	**88,125**	88,60	87,50	88,10	87,70	88,80	88,20

Çoğunluk oylamasından yararlanan önerdiğimiz özgün model, olumsuz sahte yorumlar içeren veri kümesinde 5-katlı çapraz geçerleme ile test edilmiştir. Çoğunluk oylaması test sonuçları Tablo 5-12'da gösterilmiştir.

Tablo 5-12: Çoğunluk Oylaması test sonuçları

	Gerçek			Sahte			
D	**P %**	**R %**	**F %**	**P %**	**R %**	**F %**	**Seçilen Sınıflayıcılar ile Parametreleri**
86,875	86,40	87,50	87,00	87,30	86,30	86,80	LibSVM -C 90,0 SMO supportVector,PolyKernel RandomForest -I 380 -K 0 -S 60
86,875	86,10	88,00	87,00	87,70	85,80	86,70	LibLINEAR -C 120,0 SMO supportVector,PolyKernel RandomForest -I 380 -K 0 -S 60
87,000	86,50	87,80	87,10	87,60	86,30	86,90	LibLINEAR -C 120,0 SMO supportVector,PolyKernel J48 -C 0,1
86,500	85,8	87,5	86,6	87,2	85,5	86,4	LibSVM -C 90,0 SMO supportVector,PolyKernel J48 -C 0,1
87,250	86,2	88,8	87,4	88,4	87,8	87,1	LibSVM -C 70,0 SMO - PolyKernel LibLINEAR 110,0
87,250	86	89	87,5	88,6	85,5	87	LibSVM -C 90,0 SMO supportVector,PolyKernel LibLINEAR -C 120,0
87,125	9	83,5	86,6	84,6	90,8	87,6	LibSVM -C 90,0 SMO supportVector,PolyKernel LibLINEAR -C 120,0 RandomForest -I 380, -S 60 NaiveBayesMultinomial
88,000	88,6	87,3	87,9	87,4	88,8	88,1	LibLINEAR -C 100,0 LibSVM -C 90,0 SMO supportVector,PolyKernel RandomForest -I 380 -K 0 -S 60 J40 -C 0,1
88,125	88,6	87,5	88,1	87,7	88,8	88,2	LibLINEAR -C 120,0 LibSVM -C 90,0 SMO supportVector,PolyKernel RandomForest -I 380 -K 0 -S 60 J48 -C 0,1

5.2.9. Oylama Algoritmasının İstatistiksel Karşılaştırılması

Elde ettiğimiz sonuç ve Ott ve arkadaşlarının sonucu arasında istatistiksel olarak anlamlı bir fark olup olmadığını anlamak üzere Eşli T-Test (Paired T-Test) [59] yöntemini kullandık.

Ott ve arkadaşlarının kullandıkları sınıflayıcı ile önerdiğimiz çoğunluk oylaması sınıflayıcıları ayrı ayrı karşılaştırılmıştır. İstatistiksel karşılaştırma değerleri aşağıdaki Tablo 5-13'de gösterilmiştir.

Tablo 5-13: Farklı modellerin istatistiksel karşılaştırması

Sınırlayıcıca	**İstatistiksel Değer**
Ott ve Arkadaşları DVM %86,0	85,09
Önerdiğimiz Model (Oylama): 88,125 LibLINEAR -C 120,0 LibSVM -C 90,0 SMO - PolyKernel RandomForest -I 380 -S 60 J48 -C 0,1	**86,91 v**
İstatistiksel karşılaştırmada, "v" karakteri daha iyi bir sonuç olduğunu göstermektedir,	

Tablo 5-13'de görüldüğü üzere önerdiğimiz 5 sınıflayıcı içeren modelin sunduğu performans, Ott ve arkadaşlarının [27] sunduğu performanstan istatistiksel olarak daha anlamlıdır.

6. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu tez çalışmasında çoklu sınıflayıcılı sistemleri kullanarak olumsuz aldatıcı tüketici yorumlarının tespiti gerçekleştirilmiştir. Oylama algoritmasının kombinasyon kuralı olarak çoğunluk oylaması (majority voting) kuralı kullanılmıştır. Bu sayede çok sayıda sınıflayıcının problemin farklı bölümlerini çözmesini sağlayarak, toplamda doğruluk oranı daha yüksek bir model sunulmuştur.

Sahte yorum tespiti alanında çok az çalışma olduğu, bu konunun zor ve halen çalışmaya açık bir alan olduğu görülmüştür. Sahte yorum tespiti alanında Ott ve arkadaşlarının yaptığı çalışmalar [24, 26, 27] bu alandaki öncü çalışmalar olarak değerlendirilmiştir. Ott ve arkadaşlarının DVM kullanarak [27] ulaştıkları sınıflandırma performansını iyileştirmek üzere tez kapsamında çok sayıda test ve analiz gerçekleştirilmiştir. Bu amaçla çalışmalarında kullandıkları negatif yorumlar veri kümesi bu çalışma için de referans başvuru kaynağı olarak kullanılmıştır.

Veri kümesinin sayısal ortamda işlenmesi ve sınıflandırılabilmesi için WEKA [59] veri madenciliği yazılımından yararlanılmıştır. WEKA yazılımında veri kümesinin işlenebilmesi için veri kümesinin "arff" dosya formatına dönüştürülmesi gerekmiştir. Ham metinlerden oluşan veri kümesini "arff" dosya formatına dönüştürmek amacıyla "Metin Sınıflandırma için ARFF Oluşturucu" adı verilen program [60] kullanılmıştır. Bu program kullanılarak, veri kümesinde bulunan her bir metin için bu metin içindeki kelimelerin frekans ve olasılık değerleri hesaplanmış ve tüm bilgiler "arff" uzantılı bir dosya içerisinde kaydedilmiştir. Bu dönüşüm vasıtasıyla, ham verinin Weka ortamındaki çok sayıda makine öğrenmesi yöntemiyle analiz edilebilmesi mümkün olabilmiştir. Eğer her bir makine öğrenmesi algoritmasını en baştan gerçeklemeye çalışsaydık, bu tez süresince bu tür performansı yüksek yeni bir model öneremezdik. Bu açıdan, ham veriyi "arff" formatına dönüştürmemizin yapılan çalışma açısından önemli kazanımlar kattığı değerlendirilmektedir. Makine öğrenmesi alanında bu şekilde farklı veri formatlarıyla çalışacak araştırmacılar için veri kümesini "arff" formatına taşımak zaman alıcı olsa da, ileriki çalışmalar için önemli faydalar sağlayacağı görülmüştür ve araştırmacılara tavsiye edilmektedir.

Oluşturulan "arff" uzantılı veri kümesi, WEKA programında çok sayıda sınıflayıcı kullanılarak ayrı ayrı test edilmiştir. Bu analizler öncesinde, Ott ve arkadaşlarının [27]

çalışmalarında elde etmiş oldukları %86 doğruluk oranını yeniden kendi ortamımızda elde edilebilmek için bir dizi analizler geçekleştirilmiştir. Özellikle modellerindeki maliyet (cost) parametresi, yayınlarında verilmediğinden maliyet değerini bulmak için çok sayıda test gerçekleştirmemiz gerekti. Maliyet parametresi 60,5 yapıldığında %86 doğruluk oranına ulaşmış olduk ve bu modeli kendi önereceğimiz özgün yöntemle karşılaştırabilir duruma geldik. Bu noktada, bilgisayar mühendisliği ve bilgisayar bilimlerinde deneylerin tekrarlanabilirliğinin sağlanabilmesi için ortaya konulan modellerin tüm parametrelerinin açık şekilde yayınlarda verilmesinin önemi bir kez daha görülmüştür. Araştırmacılara yayınlarını hazırlarken bu tür hususları da göz ardı etmemesi gerektiği hususunda önerilerde bulunulmaktadır.

Deney tekrarlanmasına yönelik çalışmaların ardından, diğer sınıflayıcılar ve farklı model parametreleri kullanılarak çok sayıda test gerçekleştirilmiş olup en yüksek doğruluk değerlerini veren parametreler tespit edilmiştir. İncelediğimiz yöntemlerde; Naive Bayes %65,125, SMO %84,25, Random Forest %84,750, J48 %68,75, Adaboost %85,25 ve DVM ile %87,13 doğruluk değeri sunmuştur.

Her bir sınıflayıcı için tespit edilen en yüksek doğruluk değerlerini veren parametreler kullanılarak, oluşturduğumuz çoklu sınıflayıcı modelimizde, sınıflayıcılar farklı kombinasyonlarda incelenerek farklı testler gerçekleştirilmiştir. İncelediğimiz çoklu sınıflayıcı modellerde, oylama algoritmasında kullanılan üçlü ve beşli sınıflayıcı kombinasyonlarından elde edilen performans sonuçlarının oldukça yüksek olduğu tespit edilmiştir. Üçlü kombinasyonlarda en yüksek %87,250 doğruluk değeri tespit edilmişken, beşli kombinasyonlarda en yüksek %88,125 doğruluk değerine ulaşılmıştır. Çalışmalarımızda en yüksek doğruluk oranını veren sınıflayıcı kombinasyonu; 5 sınıflayıcıdan oluşmuş olup bu sınıflayıcılar libLINEAR (maliyet parametresi:120), libSVM (maliyet parametresi:90,0), SMO (PolyKernel), RandomForest (ağaç sayısı: 380, çekirdek sayısı:60) ve J48 (maliyet parametresi: 0,1) olarak sıralanabilir.

Modelimizden elde edilen performans sonuçlarının istatistiksel olarak anlamlı bir fark oluşturup oluşturmadığını anlamak üzere, Eşli T-Test (Paired T-Test) [59] yönteminden yararlandık. Eşli test sonucunda, çoklu sınıflayıcı modelimizde elde edilen sonuçların Ott ve arkadaşlarının [27] elde ettiği sonuçtan istatistiksel olarak daha anlamlı olduğu tespit edilmiştir. Makine öğrenmesi alanında çalışacak araştırmacılara, performans analizi gerçekleştirirken bu tür istatistiksel yöntemleri göz ardı etmemesi önerilmekte ve bu

yönde ek analizleri sağlayarak gerçekten önerilen modelin daha iyi olup olmadığı anlaşılmaya çalışılmalıdır. Birçok çalışmada, bu tür istatistiksel kontrollerin yapılmadığı da tez sırasınca gözlemlenmiş ve tez içerisine bu tür ek analizlerin entegre edilmesine karar verilmiştir.

Gelecek çalışmalarda daha farklı sınıflayıcı kombinasyonu ve çok farklı sayıda parametrelerle yeni testler yapılıp performans arttırılabilir. Ayrıca, genetik algoritmalardan yararlanarak ortaya konulan modelin parametrelerinin optimizasyonunun gerçekleştirilmesi yönünde de ek çalışmalar gerçekleştirilerek performans daha yüksek seviyelere taşınabilir. Bu çalışma sayesinde, üzerinde çalışılan problem için çoklu sınıflayıcılı sistemlerin performansı arttırabildiği gösterilmiştir. Türk dili için sahte yorumlar içeren bir veri kümesi henüz mevcut olmadığından dolayı, Ott ve arkadaşlarının [27] kullanmış oldukları veri kümesinden yararlandık. Bu sayede, elde edilen modelin performansını doğrudan ilgili çalışmayla kıyaslama imkânı bulduk. Gelecek çalışmalarda Türk dili için farklı web sayfalarındaki bilgiler kullanılarak sahte yorum kümesi hazırlanabilir ve bu veri kümesi üzerinde sınıflandırma çalışmaları gerçekleştirilebilir. Sadece tüketici ürünlerindeki yorumların değil, yaygın olarak kullanılmakta olan sosyal ağ (facebook, twitter vb.) metinlerinin aldatıcı olma durumları da benzer şekilde gelecek çalışmalarda incelenebilir. Bu alanda henüz çok az sayıda çalışma yapıldığından, gelişmeye açık bir alan olduğu değerlendirilmiş ve yeni araştırmacılar için alanın zorlu ve fırsatlarla dolu olduğu tespit edilmiştir.

7. REFERANSLAR

[1] J. Taylor, «Are You Buying Reviews For Google Places?» 2012. [Çevrimiçi]. http://www.localgoldmine.com/blog/reputation-management/are-you-buying-reviews-for-google-places/. [Erişim Tarihi: 29.06.2014].

[2] D. Streitfeld, «For $2 a Star, an Online Retailer Gets 5-Star Product Reviews» 2012. [Çevrimiçi]. http://www.nytimes.com/2012/01/27/technology/for-2-a-star-a-retailer-gets-5-star-reviews.html?_r=2&ref=business. [Erişim Tarihi: 29.06.2014].

[3] A. Harmon, «Amazon Glitch Unmasks War Of Reviewers» 2004. [Çevrimiçi]. http://www.nytimes.com/2004/02/14/us/amazon-glitch-unmasks-war-of-reviewers.html. [Erişim Tarihi: 29.06.2014].

[4] D. Streitfeld, «The Best Book Reviews Money Can Buy» 2012. [Çevrimiçi]. http://www.nytimes.com/2012/08/26/business/book-reviewers-for-hire-meet-a-demand-for-online-raves.html?pagewanted=1&_r=3&partner=rss&emc=rss&. [Erişim Tarihi: 29.06.2014].

[5] B. News, «Samsung probed in Taiwan over 'fake web reviews'» 2013. [Çevrimiçi]. http://www.bbc.co.uk/news/technology-22166606. [Erişim Tarihi: 29.06.2014].

[6] ABC7 News, «Woman Paid To Post Five-Star Google Feedback» 2012. [Çevrimiçi]. http://www.thedenverchannel.com/news/woman-paid-to-post-five-star-google-feedback. [Erişim Tarihi: 29.06.2014].

[7] T. M. Mitchell, Machine Learning, ISBN: 0070428077, Pittsburgh, ABD: McGraw-Hill, 1997.

[8] R. Dale, H. Moisl and H. Somers, Handbook Of Natural Language Processing, ISBN : 0824746341, New York: CRC Press, 2000.

[9] T. Dalgleish and M. J. Power, Handbook of Cognition and Emotion, ISBN: 0-471-97836-1, Midsomer Norton, Somerset, UK: John Wiley & Sons, Ltd, 1999.

[10] W. G. Parrott, Emotions in Social Psychology, ISBN: 0863776833, Philadelphia: Psychology Press, 2001.

[11] B. Liu, Sentiment Analysis and Opinion Mining, ISBN: 1608458849, Morgan & Claypool Publishers, 2012.

[12] J. Yi ve T. Nasukawa, «Sentiment Analysis:Capturing Favorability Using Natural Language Processing, ISBN: 1-58113-583-1» *The K-CAP-03, 2nd International Conference on Knowledge Capture, 70–77*, New York, ABD, 2003.

[13] Carbonell, Jaime Guillermo, Subjective Understanding: Computer Models of Belief Systems, ISBN: 0835712125, ABD: UMI Research Press, 1981.

[14] A. L. Berger, V. J. D. Pietra ve S. A. D. Pietra, «A Maximum Entropy Approach to Natural Language Processing» *Computational Linguistics*, cilt 22, no. 1, 39-71, Mart 1996.

[15] S. Argamon-Engelson, M. Koppel ve G. Avneri, «Style-based Text Categorization:What Newspaper Am I Reading?» *AAAI Technical Report WS-98-05*, 1998.

[16] P. D. Turney, «Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews» *The 40th Annual Meeting of the Association for Computational Linguistics (ACL), 417-424*, Philadelphia, ABD, 2002.

[17] B. Pang, L. Lee ve S. Vaithyanathan, «Thumbs up? Sentiment Classi¯cation using Machine Learning Techniques» *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 79-86, Philadelphia, ABD, 2002.

[18] A.-M. Popescu ve O. Etzioni, «Extracting Product Features and Opinions from Reviews,» *the conference on Human Language Technology and Empirical Methods in Natural Language Processing, 339-346*, Stroudsburg, PA, ABD, 2005.

[19] M. Hu ve B. Liu, «Mining and Summarizing Customer Reviews» *the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168-177, ISBN:1-58113-888-1* , New York, NY, ABD, 2004.

[20] K. Dave, S. Lawrence ve D. M. Pennock, «Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews» *the 12th international conference on World Wide Web, 519-528 , ISBN:1-58113-680-3*, New York, NY, ABD, 2003.

[21] N. Jindal ve B. Liu, «Review Spam Detection» *WWW '07 Proceedings of the 16th international conference on World Wide , 1189-1190, ISBN: 978-1-59593-654-7* , Kanada, 2007.

[22] A. Z. B. Broder, «On the resemblance and containment of documents» *SEQUENCES '97 Proceedings of the Compression and Complexity of Sequences 7, Sayfa:21, ISBN:0-8186-8132-2*, Salerno, 1997.

[23] N. Jindal ve B. Liu, «Opinion spam and analysis» *WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining, 219-230, ISBN: 978-1-59593-927-2*, New York, NY, ABD, 2008.

[24] M. Ott, «Deceptive Opinion Spam Corpus v1.4» [Çevrimiçi]. http://myleott.com/op_spam/. [Erişim Tarihi: 29.06.2014].

[25] M. Ott, Y. Choi, C. Cardie ve J. T. Hancock, «Finding Deceptive Opinion Spam by Any Stretch of the Imagination» *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1,309-319, ISBN: 978-1-932432-87-9*, Stroudsburg, PA, ABD, 2011.

[26] M. Ott, C. Cardie ve J. Hancock, «Estimating the Prevalence of Deception in Online Review Communities» *the 21st international conference on World Wide Web, 201-210, ISBN: 978-1-4503-1229-5*, New York, NY, ABD, 2012.

[27] M. Ott, C. Cardie ve J. T. Hancock, «Negative Deceptive Opinion Spam» *Proceedings of NAACL-HLT 2013, 497–501,* Atlanta,Georgia, 2013.

[28] R. Y. Lau, S. Liao, R. C.-W. Kwok, K. Xu, Y. Xia ve Y. Li, «Text Mining and Probabilistic Language Modeling for Online Review Spam Detection» *ACM Transactions on Management Information Systems, No. 4,* cilt Vol. 2, p. 25, Aralık 2011.

[29] K. Sharma ve K.-I. Lin, «Review Spam Detector with Rating Consistency Check» *ACMSE '13 Proceedings of the 51st ACM Southeast Conference, Article No. 34, ISBN: 978-1-4503-1901-0*, Savannah, GA, ABD., Nisan 4-6, 2013.

[30] R. Chandy ve H. Gu, «Identifying Spam in the iOS App Store» *WebQuality '12 Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality, 56-59, ISBN: 978-1-4503-1237-0*, Lyon, France, Nisan 16, 2012.

[31] A. Morales, H. Sun ve X. Yan, «Synthetic Review Spamming and Defense» *KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 1088-1096, ISBN: 978-1-4503-2174-7* , Rio de Janeiro, Brazil, Mayıs 13–17, 2013.

[32] H. Sun, A. Morale ve X. Yan, «Synthetic Review Spamming and Defense» *KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining,1088-1096, SBN: 978-1-4503-2174-7* , Chicago, Illinois, ABD, Ağustos 11–14, 2013.

[33] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos ve R. Ghosh, «Spotting Opinion Spammers using Behavioral Footprints» *KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 632-640, ISBN: 978-1-4503-2174-7*, Chicago,Illinois, ABD, Ağustos 11–14,2013.

[34] S. Xie, G. Wang, S. Lin ve P. S. Yu, «Review Spam Detection via Temporal Pattern Discovery» *KDD '12 Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 823-831, ISBN: 978-1-4503-1462-6* , Beijing, Çin, Ağustos 12–16, 2012.

[35] E. Alpaydın, Introduction to Machine Learning, London: The MIT Press, ISBN-10: 0-262-01243-X, 2010.

[36] D. R. Bellhouse, «The Reverend Thomas Bayes, FRS:A Biography to Celebrate the Tercentenary of His Birth», *Statistical Science*, Vol. 19, No. 1, Institute of Mathematical Statistics, 2004, 3-43.

[37] K. M. Leung, «Naive Bayesian Classifier», 2007, [Çevrimiçi] http://www.share-pdf.com/81fb247fa7c54680a94dc0f3a253fd85/naiveBayesianClassifier.pdf, [Erişim Tarihi: 29.06.2014].

[38] «Naive Bayes Classification» 2013. [Çevrimiçi]. http://www.mathworks.com/help/stats/naive-bayes-classification.html. [Erişim Tarihi: 29.06.2014].

[39] A. McCallum ve K. Nigam, «A Comparison of Event Models for Naive Bayes Text Classification» *AAAI Technical Report WS-98-05*, 1998.

[40] J. Han ve M. Kamber, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publishers, ISBN 1-55860-901-6, Mart 2006.

[41] B. Scholköpf, «The Kernel Trick for Distances» *Advances in Neural Information Processing Systems 13*, 301-307, MIT Press, 2001.

[42] Ş. Şadi Evren, İş Zekası ve Veri Madenciliği (Weka ile), İstanbul: Cinius Yayınları, ISBN:6051276717, Temmuz 2013.

[43] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang and C.-J. Lin , «LIBLINEAR: A Library for Large Linear Classification», *Journal of Machine Learning Research 9*, 1871-1874, 2008.

[44] K. Sümbüloğlu, «Lojistik REgresyon Analizi» 2010. [Çevrimiçi]. http://78.189.53.61/-/bs/ess/k_sumbuloglu.pdf. [Erişim Tarihi: 29.06.2014].

[45] J. C. Platt, «Fast Training of Support Vector Machines Using Sequential Minimal Optimization» *Advances in Kernel Methods - Support Vector Learning, 41-65, ISBN: 0262194163*, MIT Press, 1998.

[46] J. Schmidhuber ve T. Schaul, «Metalearning» 2010. [Çevrimiçi]. http://www.scholarpedia.org/article/Metalearning. [Erişim Tarihi: 29.06.2014].

[47] Y. Freund ve R. E. Schapire, «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting»,*Journal of computer and system sciences 55,* cilt 5, no. 1, 119-139, Ağustos 1997.

[48] «ACM SIGACT: Prizes: Gödel» [Çevrimiçi]. http://sigact.org/Prizes/Godel/. [Erişim Tarihi: 29.06.2014].

[49] J. Matas ve J. Sochman, «AdaBoost» [Çevrimiçi]. http://www.robots.ox.ac.uk/~az/lectures/cv/adaboost_matas.pdf. [Erişim Tarihi: 01.07.2014].

[50] E. Alfaro, M. Gamez ve N. Garcia, «Adabag: An R Package for Classification with Boosting and Bagging», *Journal of Statistical Software,* cilt 54, no. 2, 2013.

[51] «CVParameterSelection» [Çevrimiçi]. http://weka.wikispaces.com/Optimizing+parameters#CVParameterSelection. [Erişim Tarihi: 29.06.2014].

[52] B. Diri, «Makine Öğrenmesine Giriş» [Çevrimiçi]. https://www.ce.yildiz.edu.tr/personal/banud/file/1560/Makine_Ogrenmesi_ML-5.pdf. [Erişim Tarihi: 22 12 2013].

[53] J. R. Quinlan, C4.5: programs for machine learning, San Francisco: Morgan Kaufmann Publishers Inc., ISBN:1-55860-238-0, 1993.

[54] K. M. Leung, «Decision Trees and Decision Rules», 2007. [Çevrimiçi]. http://cis.poly.edu/~mleung/FRE7851/f07/decisionTrees.pdf. [Erişim Tarihi: 29.06.2014].

[55] «C4.5 Example 1» [Çevrimiçi]. http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/c4.5_prob1.html. [Erişim Tarihi: 29.06.2014].

[56] «TripAdvisor,» [Çevrimiçi]. https://www.tripadvisor.com/. [Erişim Tarihi: 29.06.2014].

[57] G. Wu, D. Greene, B. Smyth ve P. Cunningham, «Distortion as a Validation Criterion in the Identification of Suspicious Reviews,» *the First Workshop on Social Media Analytics, 10-13, ISBN: 978-1-4503-0217-3*, New York, NY, ABD, 2010.

[58] «Amazon Mechanical Turk» Amazon, [Çevrimiçi]. https://www.mturk.com/mturk/. [Erişim Tarihi: 29.06.2014].

[59] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann ve I. H. Witten, «The WEKA Data Mining Software: An Update» *ACM SIGKDD Explorations Newsletter ,* cilt 11, no. 1, 10-18, Haziran 2009.

[60] H. Atasoy, «Metin Sınıflandırma İçin ARFF Oluşturucu» [Çevrimiçi]. http://www.atasoyweb.net/Metin-Siniflandirma-Icin-ARFF-Olusturucu. [Erişim Tarihi: 29.06.2014].

[61] «Downloading and installing Weka» [Çevrimiçi]. http://www.cs.waikato.ac.nz/ml/weka/downloading.html. [Erişim Tarihi: 29.06.2014].

[62] D. Powers, «Evaluation: From Precision, Recall And F-Measure To Roc,Informedness, Markedness & Correlation» *Journal of Machine Learning Technologies*, cilt 2, no. 1, 37-63, 2011.