



MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
DOĞAL DİL İŞLEME DERSİ FİNAL PROJESİ

Proje Adı

Türkçe Alaycılık Tespiti

Hasan Özgür Özdemir, 23120205090

Sefa Taşdemir, 23120205078

Ders Sorumlusu

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Mayıs, 2025

İstanbul Medeniyet Üniversitesi, İstanbul

İÇİNDEKİLER

İçindekiler	1
Şekil Listesi	2
Kısaltmalar	3
Özet.....	4
1. GİRİŞ.....	4
1.1. Projenin Amacı.....	4
1.2. Projenin Kısıtları.....	4
2. İLGİLİ ÇALIŞMALAR	5
3. YÖNTEM VE UYGULAMA	5
3.1 Veri toplama	5
3.2 Ön işleme adımları	6
3.3 Özellik çıkarımı	6
3.4 Makine öğrenmesi algoritmaları	7
3.5 Test yöntemleri	8
3.6 Değerlendirme yöntemleri.....	8
4. SONUÇLAR	10
4.1 %100 Eğitim Testi Değerlendirmesi	10
4.2 %70 Eğitim %30 Test Değerlendirmesi.....	11
4.3 5 Katlı Çarpaz Doğrulama Değerlendirmesi	12
4.4 10 Katlı Çarpaz Doğrulama Değerlendirmesi	13
5. TARTIŞMA	14
7. REFERANSLAR.....	15

ŞEKİL LİSTESİ

Tablo 1 Performans değerleri (%100 eğitim).....	10
Şekil 1 Performans değerleri (%100 eğitim).....	10
Tablo 2 Performans değerleri (%70 eğitim %30 test).....	11
Şekil 2 Performans değerleri (%70 eğitim %30 test).....	11
Tablo 3 Performans değerleri (5 katlı çarpaz doğrulama)	12
Şekil 3 Performans değerleri (5-fold cross validation)	12
Tablo 4 Performans değerleri (10 katlı çarpaz doğrulama)	13
Şekil 4 Performans değerleri (10-fold cross validation)	13

KISALTMALAR

TF-IDF: Term Frequency-Inverse Document Frequency

CSV: Comma Separated Values

TDK: Türk Dil Kurumu

AdaBoost: Adaptive Boosting

SVM: Support Vector Machine

ÖZET: Sarcasm (alaycılık), kullanıldığı metnin anlamını değiştiren bir mizah unsurudur ve sosyal medyada, mizah içeriği üreten sayfalarda bununla sık sık karşılaşırız. Bu durum doğal dil işleme sistemlerinin doğru duygu analizi yapmasını oldukça zorlaştırır. Bu çalışmada alaycı ve alaycı olmayan Türkçe haber başlıkları üzerinden bir alaycılık tespiti yapılması hedeflenmiştir. Alaycı haber başlıkları Zaytung, alaycı olmayan haber başlıkları ise TRT Haber kaynaklarından toplanmıştır. Dengeli bir şekilde toplanmış verileri makine öğrenmesi algoritmalarında kullanabilmek için preprocess adımlarında Zemberek kütüphanesi ve özellik çıkarımı için TF-IDF yöntemi kullanılmıştır. Preprocess ve özellik çıkarımı sonucunda elde edilen veri, yedi farklı makine öğrenmesi algoritması ve dört farklı test yöntemiyle değerlendirilmiş Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ve F1-Ölçütü (F1-Score) sonuçları karşılaştırılmıştır. Sonuçlara göre Logistic Regression ve SVM algoritmaları en yüksek sonucu verirken, AdaBoost en düşük sonucu vermiştir.

1. GİRİŞ

Raporun bu bölümünde projenin amacından ve kısıtlarından bahsedilmiştir. Sonraki bölümde literatürdeki benzer çalışmalardan bahsedilmiştir. Yöntem ve uygulama bölümünde; veri toplama, ön işleme, özellik çıkarımı adımlarından kullanılan makine öğrenmesi algoritmaları ve test yöntemlerinden ayrıntılı olarak bahsedilmiştir ve alınan sonuçlar sütun grafiği ve tablolar eşliğinde Sonuçlar bölümünde aktarılmıştır. Tartışma bölümünde işlem adımları ve sonuçlar değerlendirilmiş ve yapılabilecek geliştirmelerden Gelecek çalışmalar bölümünde bahsedilmiştir. Son bölümde çalışmada yararlanılan kaynakların listelenmesiyle rapor sonlandırılmıştır.

1.1 Projenin Amacı

Gelişen yapay zeka ortamında özellikle doğal dil işleme alanında duygu analizi önemli bir yer kaplamaktadır fakat alaycı metinler doğru duygu analizi yapmanın önünde büyük bir engel oluşturuyor. Bu nedenle duygu analizi yapmadan önce metnin alaycı olup olmadığını tespit etmek çok önemlidir. Biz bu çalışmada farklı makine öğrenmesi algoritmalarını TRT Haber ve alaycı içerik üreten Zaytung kaynaklarından topladığımız Türkçe verilerle eğitip dört farklı test yöntemiyle değerlendirdik ve literatürde Türkçe veriler ile yapılmış çalışmalara katkı sunmayı amaçladık.

1.2 Proje Kısıtları

a. Veri Çeşitliliği: Türkçe için etiketli, geniş kapsamlı alaycı veri seti bulunmadığından dolayı etiketleme işi direkt verilerin geldiği kaynağa göre yapılmıştır (Zaytung – sarcastic, TRT – non-sarcastic). Ayriyetten veri toplama için yalnızca iki kaynak kullanıldığından dolayı veri çeşitliliği sınırlı kalmıştır.

b. Kullanılan Modeller: Klasik makine öğrenmesi algoritmaları kullanılmıştır, BERT gibi derin öğrenme modelleri çalışmaya dahil edilmemiştir.

c. Bağlam Bağımsız: Özellik çıkarımı için kullanılan TF-IDF, frekans tabanlıdır ve kelime sıklığını esas alır bu da cümlelerin bağlamında yatan alaycılığı anlamada yetersiz kalabilir.

2. İLGİLİ ÇALIŞMALAR

Bingöl ve Yıldırım (2022), İngilizce Twitter verilerini kullanarak klasik makine öğrenmesi algoritmaları ile bir alaycılık tespit çalışması yapmışlar ve Random Forest algoritması ile %94.9'luk bir Accuracy sonucu elde etmişlerdir. [1]

Akkaya (2021), Keras tabanlı bir derin öğrenme modeli kullanarak İngilizce haber başlıkları için bir sarkazm tespit çalışması yapmıştır. Model GPU hızlandırmalı olarak 3 farklı GPU üzerinden test edilmiş (GTX 1070, RTX 3060, RTX 3060 Ti) ve %86.44'lük bir doğruluk oranı elde edilmiştir. [2]

Anup ve arkadaşları (2020), Twitter verilerini kullanarak sarkazm tespiti çalışması yapmışlar. Özellik çıkarımı için Word2Vec ve makine öğrenmesi için SVM gibi algoritmalar kullanmışlar. Verileri %80 eğitim %20 test olacak şekilde ayırmışlar ve %90-%96 arasında bir doğruluk oranı elde etmişler. [3]

Bâroiu ve Trăușan-Matu (2022), son 10 yılda otomatik sarkazm tespiti alanında yapılan çalışmaları sistematik bir şekilde analiz etmişler ve çalışmalardaki veri kaynaklarını, kullanılan modelleri ve elde edilen başarı oranlarını değerlendirmişlerdir. Çalışma, genellikle Twitter verilerinin kullanıldığını ve uzak denetimli etiketleme ile tespit edilmeye çalışıldığından bahsediyor aynı zamanda bağlam ve dil farklılıklarının önemli zorluklar yaşattığını vurguluyor. [4]

3. YÖNTEM VE UYGULAMA

3.1 Veri Toplama

İki farklı haber sitesinden toplamda 6880 haber başlığı toplanıp her biri sarcastic ya da non-sarcastic olarak etiketlenmiştir. Dengeli bir veri seti oluşturulmuştur.

a. Sarkastik Veriler - Zaytung

Zaytung, Türkiye’de en çok bilinen ironi içerikli haber sitesidir. Python’da selenium kütüphanesi ile web scraping yapılmıştır. Tam 3380 haber başlığı çekilip her biri “sarcastic” olarak etiketlenmiştir.

b. Sarkastik Olmayan Veriler - TRT Haber

Alaycı olmayan veriler için resmi ve doğrulanmış haber ajansı olan TRT Haber’in resmi web sitesi kullanılmıştır. Scraping işlemi için Python’da requests ve BeautifulSoup kütüphaneleri kullanılmıştır. Tam 3500 haber başlığı çekilip her biri “non-sarcastic” olarak etiketlenmiştir.

Ayrı ayrı çekilen metinler sonrasında tek bir csv dosyasında source (zaytung-trt), text (metnin içeriği), label (sarcastic, non-sarcastic) sütunları altında rastgele bir şekilde birleştirilmiştir.

3.2 Ön İşleme Adımları

Çalışmada kullanılan TRT Haber ve Zaytung verileri kullanıcı kaynaklı içeriklerin (Twitter, Ekşi Sözlük, Reddit...) aksine gramatik açıdan daha düzgün ve düzenli olduğundan ön işleme adımları daha sade bir şekilde kurulmuştur.

a. Metin Temizliği

Noktalama işaretleri, özel karakterler, sayısal ifadeler kaldırılmıştır. Fazla boşluklar tek boşluğa indirilmiş ve tüm metin küçük harfe dönüştürülmüştür.

b. Kök Bulma

Metinler temizlendikten sonra Türkçeye özgü kök çıkarımı işlemi için Zemberek kütüphanesi ile kelimeler için en uygun kök (lemma) tespit edilmeye çalışılmış anlamlı kök bulunamayan durumda direkt olarak kök (stem) değeri kullanılmıştır.

c. Stopword Temizliği ve Kelime Uzunluğu Filtrelemesi

TF-IDF sonucunda elde edilen bilginin yoğunluğunu artırmak amaçlanmıştır. TDK ve Zemberek kütüphanesi referans alınarak anlamsal katkısı düşük kelime listesi oluşturulmuş ve metinden çıkarılmıştır. Ayrıca çok kısa (< 3 harf) ve çok uzun (> 30 harf) kelimeler de çıkarılmıştır.

3.3 Özellik Çıkarımı

Ön işleme adımlarından sonra temizlenmiş veriyi sayısal temsillere dönüştürmek için TF-IDF yöntemi kullanılmıştır.

TF-IDF

- TF (Terim Sıklığı): Bir kelimenin bir belgede ne sıklıkta görüldüğünü ölçer. Tek başına kullanımda 've' veya ingilizcede 'the' gibi kelimeler yüksek TF puanına sahip olabileceğinden dolayı ayırt edici olarak kullanılamaz
- IDF (Ters Belge Sıklığı): Birden fazla belgedeki yaygın sözcüklerin ağırlığını azaltırken nadir sözcüklerin ağırlığını artırır. Bir terimin daha az belgede görünmesi anlamlı ve belirli olma olasılığını artırır.[5]

Özellik Çıkarımında Dikkat Edilen Durumlar

- "max_features = 1000" ile en fazla bilgi taşıyan 1000 kelime seçilmiştir.
- "ngram_range(1, 2)" ile hem tek kelimeler (unigram) hem de iki kelimelik (bigram) öbekler dikkate alınmıştır.
- "min_df = 5" ile sadece 5 veya daha fazla belgede geçen kelimeler dikkate alınmıştır. Bu şekilde, çok nadir görülen kelimeleri dışlanmıştır.
- "max_df = 0.8" ile bir kelime tüm belgelerin en az %80'inde geçiyorsa, ayırt edici bir kelime olmadığından dikkate alınmaz.

3.4 Makine Öğrenmesi Algoritmaları

Çalışmada alayıcı metinleri ayırt edebilmek için yedi farklı makine öğrenmesi algoritması kullanılmış ve bunlar kendi aralarında kıyaslanmıştır. Kullanılan algoritmalar aşağıdaki gibidir:

a. Naive Bayes

Temeli Bayes Teoremi'ne dayanmaktadır. Genelde yüksek boyutlu metin sınıflandırmalarında kullanılır. Özelliklerin birbirinden bağımsız olduğunu varsayar.[6]

b. Logistic Regression

İki veri faktörü arasındaki ilişkiyi analiz eden istatistiksel bir algoritmadır. Girdileri bağımsız değişkenler olarak alır ve 0 ile 1 arasında bir olasılık değeri üretir. Bir örneğin bir sınıfa ait olup olmadığını tahmin etmek için kullanılan gözetimli bir makine öğrenme algoritmasıdır.[7]

c. Decision Tree (J48)

C4.5 algoritması bilgi teorisine dayalı karar ağaçları üreten bir sınıflandırma algoritmasıdır.[8]

d. Random Forest

Belirli bir veri kümesinin çeşitli alt kümelerinde birden fazla karar ağacı içeren ve bu veri kümesinin tahmin edilen doğruluğunu artırmak için ortalamasını alan bir sınıflandırıcıdır.[9]

e. AdaBoost

Öğrenenlerin sırayla büyümesi ilkesine göre çalışır. İlk öğrenen hariç, sonraki her öğrenen daha önce yetiştirilen öğrenenlerden büyütülür. Kısaca zayıf öğrenenler güçlü öğrenenlere dönüştürülür.[10]

f. SVM

SVM, veri noktalarını farklı sınıflara ayırmak için N boyutlu bir uzayda optimum hiper düzlemi bulmayı amaçlar. Algoritma, farklı sınıfların en yakın noktaları arasındaki marjı en üst düzeye çıkarır.[11]

g. Extra Trees

Veri kümesinin çeşitli alt örneklerine bir dizi rastgele karar ağacı uygulayan ve tahmin doğruluğunu iyileştirmek ve aşırı uyumu kontrol etmek için ortalama kullanan bir meta tahmin edici uygular [12]. Random Forest'a benzer fakat her ağaç için düğüm bölünmesini rastgele seçer.

3.5 Test Yöntemleri

a. %100 Eğitim Verisi ile Test

Model, tüm veriyle eğitilir ve yine aynı verilerle test edilir. Modelin ezberleme eğilimini görmek amacıyla yapılır, gerçek başarıyı göstermez. Başarı oranı genelde yüksektir.

b. %70 Eğitim %30 Test Spliti

Veri kümesinin %70'i eğitim ve %30'u test verisi olarak ikiye ayrılır. Model eğitim verisi ile eğitilir ve hiç görmediği %30'luk veri ile test edilir. Veriyi ikiye ayırma işlemi rastgele yapıldığı için yapılan ayırmaya göre sonuç değişebilir.

c. 5-Fold Cross Validation (5 Katlı Çarpaz Doğrulama)

Veri 5 eşit parçaya bölünür, 4 parçası eğitim 1 parçası test için kullanılır. Aynı işlem 5 kez tekrarlanır her bir parça test için kullanılmış olur. Çıkan beş sonucun ortalaması alınır. Eğitim ve test ayrımı her partta farklı olacağı için daha güvenilir sonuç verir.

d. 10-Fold Cross Validation (10 Katlı Çarpaz Doğrulama)

5-Fold Cross Validation'ın aynısı fakat bu sefer 10 parça kullanılır. Her parça daha küçük olduğundan daha hassas sonuç sunar.

3.6 Değerlendirme Yöntemleri

Modellerin başarı oranlarını objektif bir şekilde değerlendirebilmek için çeşitli değerlendirme yöntemleri kullanılması gerekir. Biz bu çalışmada Accuracy, Precision, Recall ve F1-Score değerlendirme yöntemlerini tercih ettik.

Ön Bilgi

- TP: Doğru Pozitif (sarkastik olanı doğru tahmin etmesi)
- TN: Doğru Negatif (sarkastik olmayanı doğru tahmin etmesi)
- FP: Yanlış Pozitif (sarkastik olmayan veriye sarkastik diyerek yanlış tahmin etmesi)
- FN: Yanlış Negatif (sarkastik olan veriye sarkastik değil diyerek yanlış tahmin etmesi)

a. Accuracy (Doğruluk Oranı)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Modelin doğru sınıflandırdığı örneklerin, toplam örnek sayısına oranını verir.
- Genel başarı oranını verir fakat veri dengesizse sonuçlar yanıltıcı olabilir. Örneğin 950 non-sarcastic, 50 sarcastic toplamda 1000 verimiz var diyelim model hepsine non-sarcastic dediğinde %95'lik bir doğruluk oranı çıkacak fakat aslında hiçbir sarcastic veriyi doğru tahmin edememiş oldu.

b. Precision (Kesinlik)

$$Precision = \frac{TP}{TP + FP}$$

- Modelin pozitif tahmin ettiği örneklerin gerçekten pozitif olma oranını verir. Yüksek kesinlik oranı, modelin gereksiz pozitif tahmin yapmadığını gösterir

c. Recall (Duyarlılık)

$$Recall = \frac{TP}{TP + FN}$$

- Gerçek pozitif örneklerin, model tarafından doğru şekilde pozitif olarak tahmin edilme oranını verir. Recall değerinin yüksek olması sarkastik verileri doğru bir şekilde bulabildiğini gösterir

d. F1-Score (F-Ölçütü)

$$F1\ Score = 2x \frac{Precision \times Recall}{Precision + Recall}$$

- Precision ve Recall değerlerinin harmonik ortalamasıdır. İki metrik arasında denge kurar

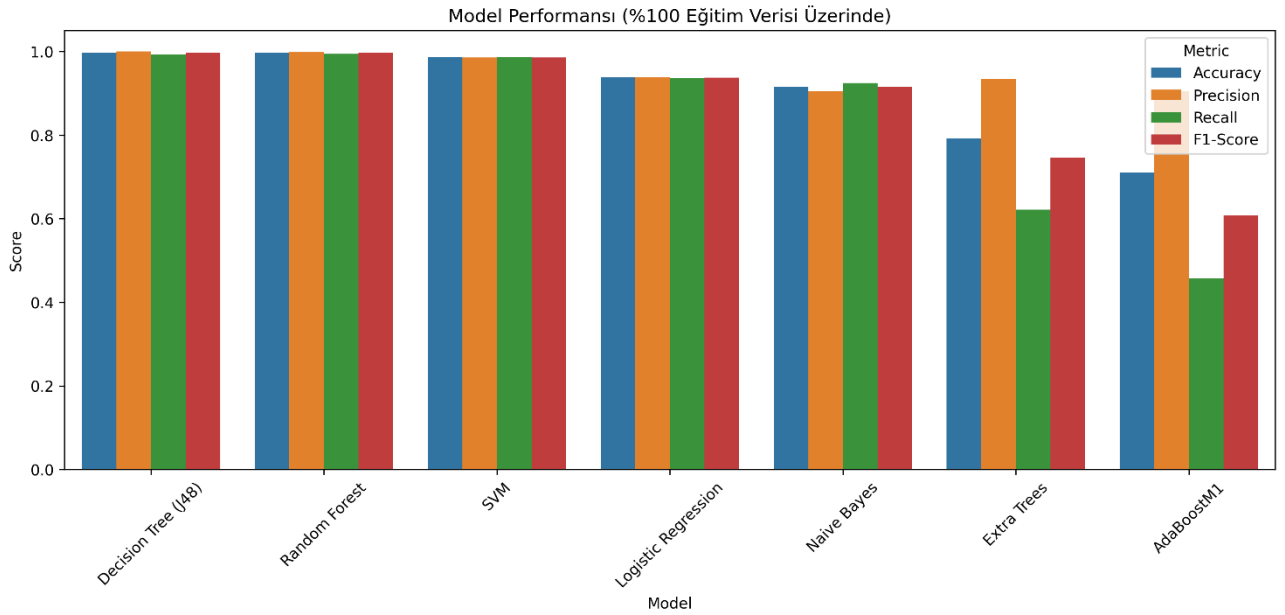
4. SONUÇLAR

4.1 %100 Eğitim Testi Değerlendirmesi

Bu senaryoda model tüm veriyle eğitilmiş ardından aynı veriyle test edilmiştir. Bu, modelin ezberleme eğilimini göstermek için kullanılmıştır.

	Naive Bayes	Logistic Regression	J48	Random Forest	AdaBoost	SVM	Extra Trees
Accuracy	0.915	0.938	0.996	0.996	0.709	0.986	0.792
Precision	0.905	0.938	1.000	0.998	0.904	0.985	0.934
Recall	0.925	0.936	0.992	0.994	0.457	0.986	0.621
F1-Score	0.915	0.937	0.996	0.996	0.607	0.986	0.746

Tablo 1. Performans değerleri (%100 eğitim)



Şekil 1 Performans değerleri (%100 eğitim)

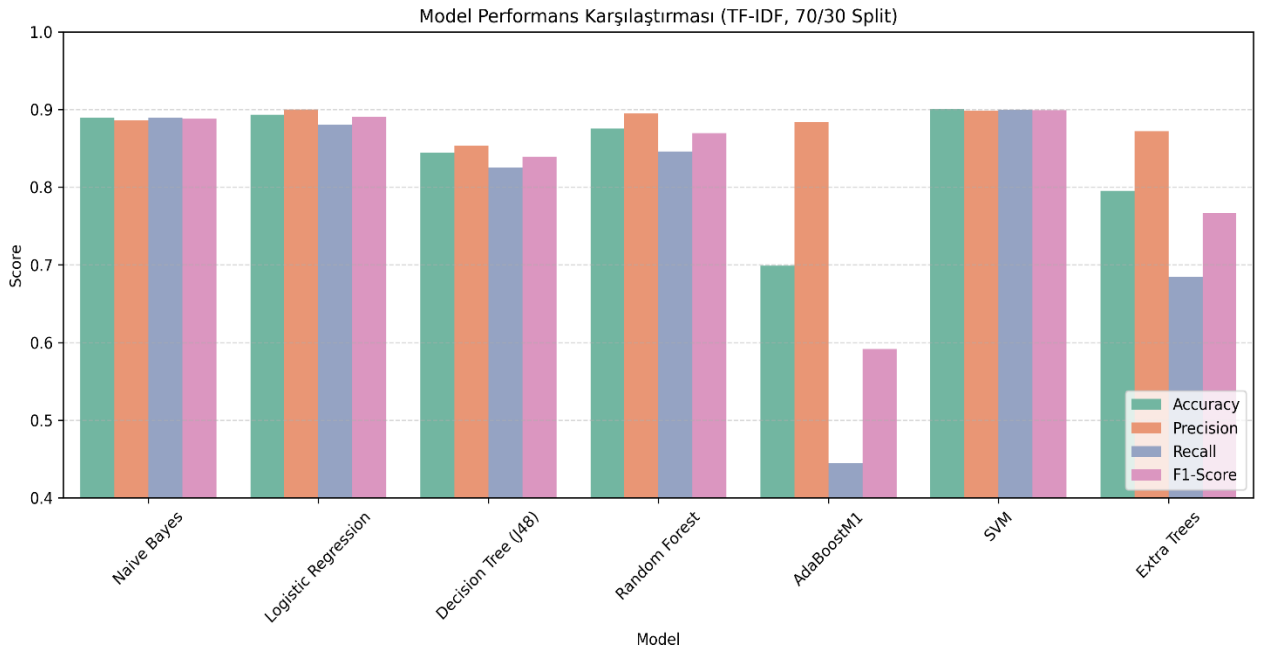
- %100 eğitim verisiyle yapılan değerlendirmede en yüksek sonucu %99 ile J48 ve Random Forest almıştır

4.2 %70 Eğitim %30 Test Değerlendirmesi

Klasik veri bölme stratejisi olan bu yöntemde veri %70 eğitim ve %30 test verisi olarak ikiye bölünüp model üzerinde denenmiştir.

	Naive Bayes	Logistic Regression	J48	Random Forest	AdaBoost	SVM	Extra Trees
Accuracy	0.889	0.893	0.844	0.875	0.698	0.900	0.7995
Precision	0.886	0.900	0.853	0.894	0.884	0.898	0.871
Recall	0.889	0.880	0.825	0.846	0.444	0.899	0.684
F1-Score	0.887	0.890	0.839	0.869	0.591	0.898	0.766

Tablo 2. Performans değerleri (%70 eğitim %30 test)



Şekil 2 Performans değerleri (%70 eğitim %30 test)

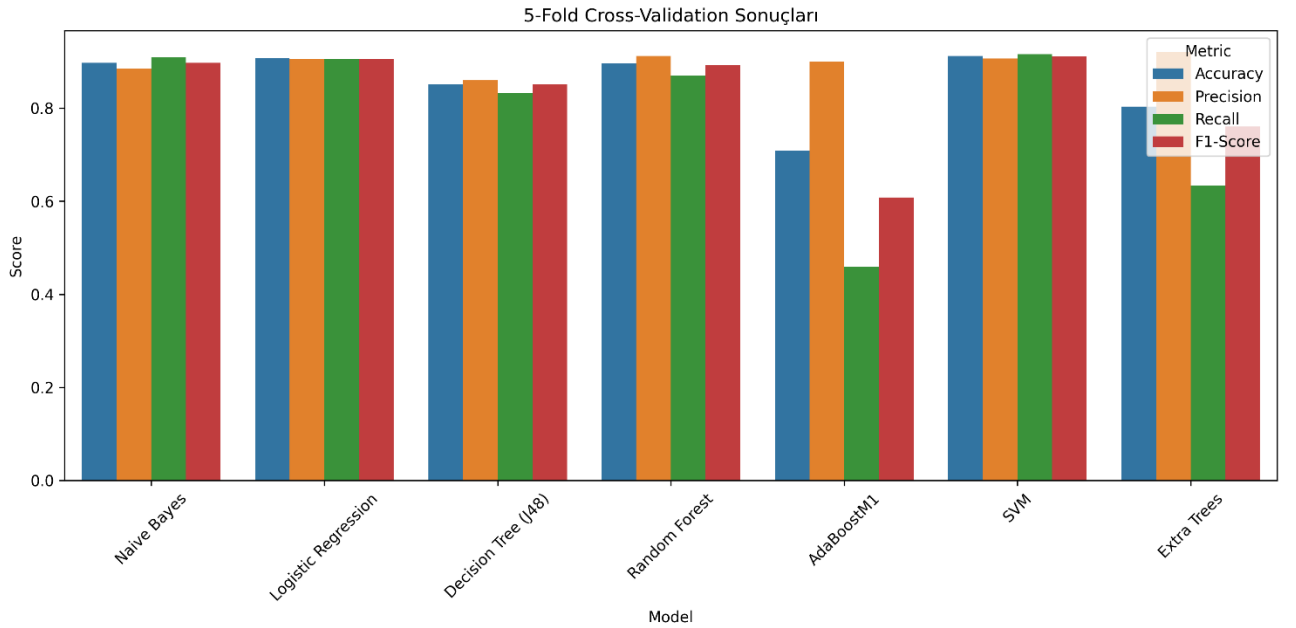
- %70 eğitim, %30 test ayrımı ile yapılan testte en yüksek doğruluk oranını %90 ile SVM algoritması göstermiştir. AdaBoost algoritması düşük bir başarı oranı göstermiştir.

4.3 5 Katlı Çarpraz Doğrulama Değerlendirmesi

Veri 5 eşit parçaya bölünür, 4 parçası eğitim 1 parçası test için kullanılır. Aynı işlem 5 kez tekrarlanır her bir parça test için kullanılmış olur. Çıkan beş sonucun ortalaması alınır.

	Naive Bayes	Logistic Regression	J48	Random Forest	AdaBoost	SVM	Extra Trees
Accuracy	0.897	0.906	0.851	0.895	0.709	0.912	0.803
Precision	0.885	0.905	0.859	0.912	0.900	0.906	0.920
Recall	0.909	0.905	0.832	0.869	0.459	0.915	0.633
F1-Score	0.897	0.905	0.851	0.892	0.608	0.911	0.760

Tablo 3. Performans değerleri (5 katlı çarpraz doğrulama)



Şekil 3 Performans değerleri (5-fold cross validation)

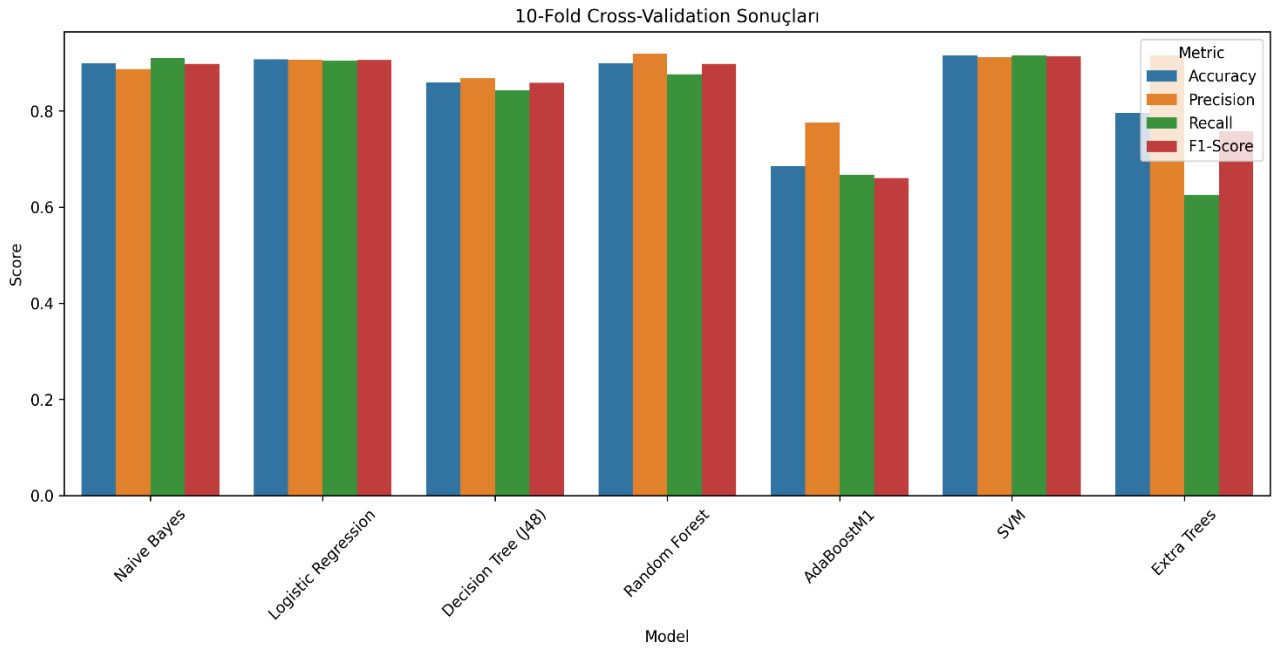
- Değerlendirme sonucunda SVM algoritması, %91 doğruluk ve F1-Ölçütü değerlerine ulaşmıştır.

4.4 10 Katlı Çarpaz Doğrulama Değerlendirmesi

5-Fold Cross Validation'ın aynısı fakat bu sefer 10 parça kullanılır.

	Naive Bayes	Logistic Regression	J48	Random Forest	AdaBoost	SVM	Extra Trees
Accuracy	0.898	0.907	0.859	0.900	0.685	0.915	0.796
Precision	0.887	0.906	0.867	0.919	0.776	0.912	0.776
Recall	0.909	0.905	0.842	0.876	0.666	0.915	0.666
F1-Score	0.898	0.905	0.858	0.897	0.660	0.913	0.660

Tablo 4. Performans değerleri (10 katlı çarpaz doğrulama)



Şekil 4 Performans değerleri (10-fold cross validation)

- 10 Katlı çarpaz doğrulama testi sonucunda da SVM algoritması %91.5 ile en yüksek başarı oranını göstermiştir.

5. TARTIŞMA

Çalışma sonucunda beklenenden yüksek başarı oranları elde edilmiştir özellikle SVM ve Logistic Regression (%90+ doğruluk) modelleri yüksek başarı göstermiştir. Buna neden olarak cümlelerin kısa ve net ifadeler içermesi, Zaytung ve TRT Haber arasında dil farklılığının fazla olması düşünülmüştür. Bu nedenle daha karmaşık metinlerde örneğin kullanıcı tarafından gönderilmiş bir tweet için performans düşük olabilir. Random Forest ve Decision Tree gibi algoritmalar farklı çalışmalarda diğer algoritmalara göre daha yüksek başarı oranı gösterirken bu çalışmada nispeten daha az başarı göstermişlerdir. Türkçe’de geniş kapsamlı, etiketli veri kısıtlılığı olması çalışma alanımızı daraltmıştır. Gelecekte yapılacak çalışmalarda veri seti oluşturmak için daha fazla vakit ayırıp çeşitli ve daha fazla sayıda veriyle çalışmak alınan sonuçları daha genelleştirilebilir yapacaktır. Aynı zamanda klasik makine öğrenmesi algoritmaları yerine derin öğrenme metodları kullanmak da alınan sonuçlara oldukça pozitif etkisi olacaktır.

6. REFERANSLAR

- [1] Bingöl, H., & Yıldırım, M. (2022). Sarcasm detection in online social networks using machine learning methods. *NATURENGS, MTU Journal of Engineering and Natural Sciences*, 3(1), 42–53. <https://doi.org/10.46572/naturengs.1100358>
- [2] Akkaya, A. E. (2021). Sarcastic text detection using Keras. *Journal of Computer Science, IDAP-2021*(Special Issue), 126–131. <https://doi.org/10.53070/bbd.990890>
- [3] Anup, A., Shruthi, G., Shruthi, H. R., Upadhyaya, M., Ray, A. P., & Manjunath, T. C. (2021). Sarcasm detection in natural language processing. *Materials Today: Proceedings*, 37, 3324–3331. <https://doi.org/10.1016/j.matpr.2020.09.124>
- [4] Băroiu, A.-C., & Trăuşan-Matu, Ş. (2022). Automatic sarcasm detection: Systematic literature review. *Information*, 13(8), 399. <https://doi.org/10.3390/info13080399>
- [5] GeeksforGeeks. (2025, Şubat 7). *Understanding TF-IDF (Term Frequency-Inverse Document Frequency)*. <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>
- [6] GeeksforGeeks. (2025, May 21). *Naive Bayes Classifiers*. <https://www.geeksforgeeks.org/naive-bayes-classifiers/>
- [7] GeeksforGeeks. (2025, Şubat 3). *Understanding Logistic Regression*. <https://www.geeksforgeeks.org/understanding-logistic-regression/>
- [8] Khanna, N. (2021, Ağustos 18). *J48 Classification (C4.5 Algorithm) in a Nutshell*. Medium. <https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e>
- [9] Turing.com. (2022, Haziran 28). *Random Forest Algorithm - How It Works and Why It Is So Effective*. <https://www.turing.com/kb/random-forest-algorithm>
- [10] Great Learning. (2024, Eylül 2). *The Ultimate Guide to AdaBoost Algorithm*. <https://www.mygreatlearning.com/blog/adaboost-algorithm/>
- [11] GeeksforGeeks. (2025, Ocak 27). *Support Vector Machine (SVM) Algorithm*. <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>
- [12] scikit-learn developers. (2025). *ExtraTreesClassifier*. scikit-learn 1.6.1 documentation. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html>