NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2025; 14(1), 263-270
Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences
Araştırma makalesi / Research article
www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Farklı kodlama tekniklerinin KNN algoritmasının mantar sınıflandırma performansı üzerindeki etkisi

Effect of different encoding techniques on the mushroom classification performance of KNN algorithm

Kadir İleri [1,*]

[1] *Bandırma Onyedi Eylül Üniversitesi, Elektrik- Elektronik Mühendisliği Bölümü, 10200, Balıkesir, Türkiye*

Öz

Bu çalışmada, mantarların zehirli veya yenilebilir olarak sınıflandırılmasında farklı kodlama tekniklerinin K-En Yakın Komşu (KNN) algoritması üzerindeki etkisi araştırılmıştır. Etiket kodlama, one-hot kodlama, frekans kodlama, hash kodlama ve hedef kodlama gibi çeşitli kodlama teknikleri kullanılarak, çoğunlukla kategorik özellikler içiren bir veri setindeki kategorik özellikler sayısal verilere dönüştürülmüştür. Modelin performansı doğruluk, kesinlik, duyarlılık ve f1-skoru gibi metriklerle değerlendirilmiştir. Sonuçlar, frekans kodlamanın k=1 durumunda en iyi performansı sergilediğini, hedef kodlamanın ise k=7 durumunda en düşük performansı gösterdiğini ortaya koymuştur. Çalışmanın bulguları, kategorik veri dönüşümünün KNN modeli üzerindeki etkilerini anlamak ve daha doğru sınıflandırma sonuçları elde etmek için önemli ipuçları sunmaktadır.

Anahtar kelimeler: KNN sınıflandırıcısı, Kategorik veri, Etiket kodlama, One-hot kodlama, Frekans kodlama

Abstract

In this study, the effects of different encoding techniques on the K-Nearest Neighbors (KNN) algorithm in the classification of mushrooms as poisonous or edible were investigated. Various encoding techniques such as label encoding, one-hot encoding, frequency encoding, hash encoding, and target encoding were used to convert categorical features in a dataset, which mostly contains categorical features, into numerical data. The performance of the model was evaluated using metrics such as accuracy, precision, recall, and f1-score. The results revealed that frequency encoding showed the best performance at k=1, while target encoding showed the lowest performance at k=7. The findings of the study provide significant insights into understanding the impact of categorical data transformation on the KNN model and achieving more accurate classification results.

Keywords: KNN classifier, Categorical data, Label encoding, One-hot encoding, Frequency encoding

1 Giriş

Makine öğrenimi ve veri bilimi alanları, giderek artan veri miktarı ve bu verilerin işlenmesi gereksinimiyle birlikte hızla gelişmektedir. Özellikle kategorik verilerin etkili bir şekilde kodlanması, makine öğrenimi modellerinin performansını önemli ölçüde etkileyebilmektedir [1]. Kategorik veriler genellikle sınırlı sayıda farklı değere sahip olup, bu değerler sıralı ya da sırasız olabilir. Bu tür veriler, doğrudan makine öğrenimi algoritmalarına giriş olarak verilemez ve bu nedenle, uygun şekilde sayısal değerlere dönüştürülmeleri gerekmektedir [2].

Bu çalışma, çeşitli kodlama tekniklerinin KNN algoritması kullanılarak gerçekleştirilen bir sınıflandırma problemi üzerindeki performansını analiz etmeyi amaçlamaktadır. KNN algoritması, basitliği ve açıklana bilirliği nedeniyle sıklıkla tercih edilen bir denetimli öğrenme yöntemidir [3]. Ancak, KNN'nin performansı, kullanılan özelliklerin temsiline büyük ölçüde bağımlıdır. Bu nedenle, kategorik verilerin doğru bir şekilde kodlanması, KNN algoritmasının başarısını artırmada kritik bir rol oynar [4].

Bu çalışmada, yukarıda bahsedilen kodlama tekniklerinin, mantar sınıflandırma problemi üzerinde KNN algoritması ile performansları karşılaştırılmıştır. Mantar verileri, biyolojik ve ekolojik araştırmalarda önemli bir rol oynamakta olup, bu tür verilerin zehirli veya yenilebilir olarak sınıflandırılması insan sağlığı için büyük bir öneme sahiptir. Bu doğrultuda gerçekleştirilen çalışma, farklı k değerlerinde KNN algoritmasının doğruluk ve hata oranlarını analiz ederek, hangi kodlama tekniklerinin daha etkili olduğunu analize etmiştir.

Makalenin geri kalanı şu şekilde tasarlanmıştır. Bölüm 2 literatür taramasını içermektedir. Bölüm 3'te çalışmada kullanılan veri seti tanıtılmıştır. Bölüm 4'te kullanılan metotlar açıklanmıştır. Bölüm 5'te analizler sonucunda elde edilen sonuçlar paylaşılmış ve yorumlanmıştır. Son olarak, Bölüm 6 sonuç kısmını içermektedir.

2 Literatür taraması

Literatürde, kategorik veri dönüşümü için çeşitli yöntemler önerilmiştir. En yaygın olarak kullanılan tekniklerden bazıları etiket kodlama, one-hot kodlama, frekans kodlama, hash kodlama ve hedef kodlamadır [5-14]. Bu tekniklerin her biri, verilerin belirli bir şekilde sayısal

* Sorumlu yazar / Corresponding author, e-posta / e-mail: kileri@bandirma.edu.tr (K. İleri)
Geliş / Recieved: 12.07.2024 Kabul / Accepted: 16.12.2024 Yayımlanma / Published: 15.01.2025
doi: 10.28948/ngumuh.1515387

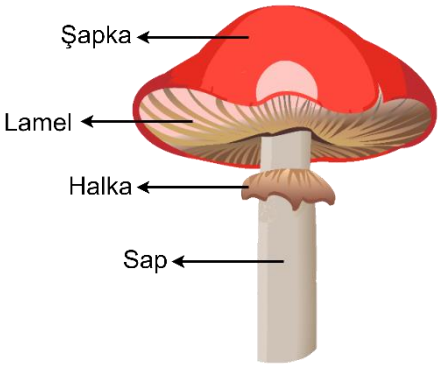
değerlere dönüştürülmesini sağlar ve her birinin kendi avantajları ve dezavantajları bulunmaktadır. Örneğin, etiket kodlama basit ve hızlıdır, ancak sıralı olmayan verilerde yanlış ilişkiler oluşturabilir. One-hot kodlama, bu sorunu aşar ancak yüksek boyutluluk sorununa neden olabilir. Frekans kodlama, kategorilerin veri setindeki frekanslarına göre kodlanmasını sağlar ve hash kodlama, büyük veri setlerinde bellek tasarrufu sağlarken hash çakışması (aynı değerin üretilmesi durumu) da yaşanabilmektedir. Hedef kodlama ise, kategorilerin hedef değişkenle olan ilişkisine göre kodlanmasını sağlar ancak aşırı uyum riskini taşır [15].

Yapılan çalışmalardan birinde, Ping Yan [5] bozulmuş verileri tespit etmek için meta veri özelliklerini kullanmış ve etkili sınıflandırma için ağaç tabanlı bir makine öğrenme algoritması uygulamıştır. İsveçli otomotiv şirketi Veoneer'den gelen bir sorunu ele alarak, %97 sağlıklı ve %3 bozuk veriden oluşan, tamamen nominal kategorik değişkenler içeren son derece dengesiz bir veri seti kullanılmıştır. Bu kategorik değişkenleri sayısal değişkenlere dönüştürmek için one-hot kodlama tekniği kullanılmıştır. Metodoloji, veri temizleme ve veri kodlama ön işlemlerini içerir. Sınıflandırma işlemi ise rastgele orman algoritması ile yapılmış ve bu algoritmanın parametre optimizasyonu için ise rastgele arama kullanılmıştır. Sonuçlar, bu yaklaşımın sınıflandırma performansını artırdığını ve meta veri özelliklerinin bozuk verileri etkili bir şekilde tespit edebildiğini ve model için kilit özellikleri tanımlayabildiğini göstermiştir. Benzer olarak, Kartik Budholiya vd. [6] kalp hastalığını tahmin etmek için optimize edilmiş bir XGBoost sınıflandırıcısı kullanmışlardır. XGBoost modelinin hiper parametreleri, oldukça verimli bir yöntem olan Bayes optimizasyonu kullanılarak optimize edilmiştir. Veri seti olarak Cleveland kalp hastalığı veri seti kullanılmış ve bu veri setinin içerdiği kategorik özellikleri sayısal verilere dönüştürmek için one-hot kodlama tekniği kullanılmıştır. Modelin etkinliği, rastgele orman ve ekstra ağaç sınıflandırıcıları ile karşılaştırılmıştır. Modellerin performans değerlendirmesi için doğruluk, duyarlılık, özgüllük, f1-skoru ve ROC eğrisi olmak üzere toplam beş farklı değerlendirme metriği kullanılmıştır. Deneysel sonuçlar, önerilen modelin kalp hastalığı tahmininde %91.8 doğruluk elde ederek etkili sınıflandırma işlemi gerçekleştirdiğini göstermiştir. Bir diğer çalışmada, Taher Al-Shehari vd. [7] tehdit olaylarını tespit etmek için makine öğrenmesi tabanlı bir model önermişlerdir. Önerilen model, özellik ölçekleme ve one-hot kodlama ön işlemlerini içermektedir. Ayrıca, kullanılan veri setindeki dengesizlik sorunu, sentetik azınlık aşırı örnekleme tekniği (SMOTE) ile ele alınmıştır. Çalışma, kötü niyetli içeriden kişilerin bir organizasyondan ayrılmadan önce gerçekleştirdiği veri sızıntısı olaylarını tespit edebilecek en doğru sınıflandırıcıyı belirlemek için lojistik regresyon, rastgele orman, destek vektör makinesi, naive bayes, KNN ve karar ağacı makine öğrenmesi algoritmalarını karşılaştırmışlardır. CMU-CERT Insider Threat Dataset adlı veri seti üzerinde modellerin sınıflandırma başarımı test edilmiş ve elde edilen sonuçlar rastgele orman ve karar ağacı modellerin diğer modellere göre en yüksek sınıflandırma başarımı göstermişlerdir. Ayrıca, aynı veri seti üzerinde gerçekleştirilen mevcut yaklaşımları performans açısından geride bırakmıştır. Mohamed Hosni [8] ise yazılım geliştirme çabası tahmini (SDEE) veri setlerindeki kategorik verilerin kodlama teknikleriyle nasıl işlenebileceğini üzerine bir analiz gerçekleştirmiştir. Çalışmada, one-hot kodlayıcı, etiket kodlayıcı, frekans kodlayıcı ve hedef kodlayıcı dahil olmak üzere dört kodlayıcı kullanılmıştır. Model olarak ise, KNN, destek vektör makinesi, çok katmanlı algılayıcı ve karar ağacı makine öğrenmesi modelleri kullanılmıştır. Analizler, ISBG, Nasa93, Maxwell ve USP05 olmak üzere toplam dört veri seti kullanılarak gerçekleştirilmiştir. One-hot kodlayıcı ile üretilen veri setleri, KNN modeli ile en iyi sonuçları vermiş ve daha doğru tahminler yapmasını sağlamıştır. Başka bir çalışmada, Min Xuan Low vd. [9] kötü amaçlı yazılım tespitini hedeflemişlerdir. Bu kapsamda, etiket kodlama ve kanıt sayma olmak üzere iki farklı özellik mühendisliği işlemi gerçekleştirilmiştir. Daha az önemli özellikleri izole etmek için özellik seçimi uygulanmıştır. Makine öğrenmesi modelleri olarak rastgele orman, karar ağacı, KNN, destek vektör makinesi, çok katmanlı algılayıcı ve uzun kısa süreli bellek sınıflandırıcıları kullanılmıştır. Modellerin performansları, doğruluk, kesinlik, duyarlılık, f1-skoru ve kayıp gibi ölçütler ile değerlendirilmiştir. Rastgele orman ve karar ağacı modelleri, diğer modellere göre daha iyi performans göstermişlerdir. Benzer olarak, Subrata Kumar Das vd. [10] büyük veri setleri için, özellikle hasta verileri için, en uygun kodlama tekniği ve öğrenme modelini belirlemeyi amaçlamışlardır. Bulgular, bazı modellerin farklı kodlama teknikleri kullanılarak eğitildiğinde büyük veri setlerinde kötü performans gösterdiğini ve eğitim sürelerinin değişkenlik gösterdiğini ortaya koymaktadır. Lineer ayrım analizi modeli, sağlık verileri setinde ortalama eğitim süresi ile en iyi bir performansı sergilemiştir. Ayrıca, etiket kodlama tekniğinin daha düşük boyutlu verilerle daha iyi performans göstermiştir. Bir diğer çalışmada, Florian Pargent vd. [11] kategorik değişkenleri sayısal temsillere dönüştürme yöntemlerini incelemişler ve bu yöntemlerin makine öğrenmesi algoritmalarının performansı üzerindeki etkilerini değerlendirmişlerdir. Büyük ölçekli bir karşılaştırmada, farklı kodlama stratejilerini beş makine öğrenimi algoritması (lasso, rastgele orman, gradyan artırma, KNN ve destek vektör makinesi) ile regresyon, ikili ve çoklu sınıflandırma görevleri üzerinde analizler gerçekleştirilmiştir. Sonuç olarak, hedef kodlamanın, etiket kodlama ve one-hot kodlamaya göre daha iyi sonuçlar verdiği gözlemlenmiştir. Ashima Sindhu Mohanty vd. [12] ise dengesiz dağılımlı Otizm Spektrum Bozukluğu veri setini kullanarak bir sınıflandırma işlemi gerçekleştirmişlerdir. Çalışma üç aşamada gerçekleştirilmiştir. İlk aşamada, veri setindeki kategorik özellikler frekans kodlama yöntemiyle sayısal değerlere dönüştürülmüş ve ardından sayısal özellikler z-skor metodu ile standartlaştırma ön işlemine tabi tutulmuştur. İkinci aşamada, veri seti boyutu temel bileşen analizi kullanılarak azaltılmıştır. Son olarak, iki aşamada farklı makine öğrenmesi modelleri ile sınıflandırma işlemi gerçekleştirilmiştir. En yüksek sınıflandırma performansı %99 doğruluk ile destek vektör makinesi modeli ile elde edilmiştir. Yine benzer bir çalışmada, Shasha Zhang vd. [13]

koroner arter hastalığını erken aşamada doğru şekilde teşhis edebilecek bir makine öğrenmesi sistemi geliştirmişlerdir. Bu kapsamda hem XGBoost ve hem de rastgele orman algoritmalarını kullanarak sınıflandırma performanslarını karşılaştırmışlardır. Veri setindeki kategorik özellikleri sayısal değerlere dönüştürmek için frekans kodlama kullanılmıştır. Ayrıca, veri setini dengelemek için SMOTE ve ADASYN işlemleri uygulanmıştır. XGBoost algoritması, özellik oluşturma ve SMOTE ile işlenmiş veri setinde %94.7 doğruluk, değeri ile en iyi performansı göstermiştir. Ludmila B. S. Nascimento vd. [14] ise majör depresif bozukluk yaşayan çocuk ve ergenlerde depresyonla ilgili bir veri setinde, sırasız nominal kategorik özellikler için farklı kodlama yöntemlerini araştırmışlardır. Karşılaştırma sonuçları, hash kodlama tekniği ile kodlanmış veri seti ile beslenen XGBoost modelinin daha etkili olduğunu ortaya koymuştur. Yapılan çalışmalar incelendiği zaman farklı kodlama tekniklerinin farklı model ve veri setlerinde farklı sonuçlar verdiği gözlemlenmiştir. Bu çalışmada da farklı kodlama tekniklerinden hangisinin KNN algoritması üzerinde etkili sonuçlar verdiği analiz edilmektedir.

3 Problem tanımı

Bu çalışma, Dennis Wagner vd. [16] tarafından 2020 yılında oluşturulan, Sekonder Mantar Veri Seti'ni (Secondary Mushroom Dataset) [17] kullanmaktadır. Bu veri seti oluşturulurken, Jeff Schlimmer'in 1981'de katkıda bulunduğu ve 2016'da güncellediği Mantar Veri Seti'nden (Mushroom Dataset) [18] ilham alınmıştır. Veri seti, 23 farklı familyadan 173 farklı türden oluşan toplam 61.069 hipotetik mantarı içermektedir. Bu kapsamlı veri seti, mantarların yenilebilir veya zehirli olarak sınıflandırılması için tasarlanmıştır. Şekil 1 bir mantarın temel özelliklerini göstermektedir. Bu özellikler, bir mantarın zehirli ya da yenilebilir olarak sınıflandırılmasına yardımcı olur.



Şekil 1. Bir mantarın temel özellikleri

Veri seti oluşturulurken, mantarların yenile bilirliğini belirlemede kritik olan 20 farklı özellik sunulmuştur. Bu özelliklerden çok fazla kayıp değere sahip olan 5 özellik çıkartılmış ve hiçbir kayıp değer içermeyen 15 özellik kullanılmıştır. Bu özellikler arasında; şapka çapı, şapka şekli, şapka yüzeyi, şapka rengi, morarıp morarmadığı, lamel bağlantısı, lamel aralığı, lamel rengi, sap yüksekliği, sap genişliği, sap rengi, halka olup olmadığı, halka tipi, habitat ve mevsim bulunmaktadır. Her bir özellik, genel sınıflandırmaya katkıda bulunan spesifik karakteristikler sağlamaktadır. Veri setinde bulunan tüm özelliklerin kod, ad, tür ve aldığı olası değer bilgileri Tablo 1'de gösterilmiştir. Burada, F1'den F15'e kadar olan veriler özellikleri temsil ederken, C ise hedef değişkeni temsil etmektedir. Üç tane özellik (F1, F9 ve F10) sayısal değerler içerirken diğer özellikler ise kategorik veriler içermektedir. C hedef değişkeni ise e (yenilebilir) veya p (zehirli) olmak üzere iki değerden birini almaktadır. Bu durum veri setinin ikili sınıflandırmaya uygun olduğunu göstermektedir. Ayrıca, Tablo 2'de de veri setine ait birkaç örnek verilmiştir.

Tablo 1. Veri setinde bulundan tüm özelliklerin bilgileri

Kod	Ad	Veri Türü	Değer
F1	şapka çapı	sayısal	Kayan noktalı sayılar (cm cinsinden)
F2	şapka şekli	kategorik	çan=b, konik=c, tümsek=x, düz=f, batık=s, küresel=p, diğerleri=o
F3	şapka yüzeyi	kategorik	lifli=i, oluklu=g, pullu=y, düz=s, parlak=h, deri=l, ipeksi=k, yapışkan=t, kırışık=w, etli=e
F4	şapka rengi	kategorik	kahverengi=n, sarımsı=b, gri=g, yeşil=r, pembe=p, mor=u, kırmızı=e, beyaz=w, sarı=y, mavi=l, turuncu=o, siyah=k
F5	morarıp morarmadığı	kategorik	morarıyor=t, morarmıyor=f
F6	lamel bağlantısı	kategorik	bitişik=a, serbest=x, inişli=d, serbest=e, sinüat=s, gözenekli=p, yok=f
F7	lamel aralığı	kategorik	yakın=c, uzak=d, yok=f
F8	lamel rengi	kategorik	kahverengi=n, sarımsı=b, gri=g, yeşil=r, pembe=p, mor=u, kırmızı=e, beyaz=w, sarı=y, mavi=l, turuncu=o, siyah=k, yok=f
F9	sap yüksekliği	sayısal	Kayan noktalı sayılar (cm cinsinden)
F10	sap genişliği	sayısal	Kayan noktalı sayılar (mm cinsinden)
F11	sap rengi	kategorik	kahverengi=n, sarımsı=b, gri=g, yeşil=r, pembe=p, mor=u, kırmızı=e, beyaz=w, sarı=y, mavi=l, turuncu=o, siyah=k, yok=f
F12	halka olup olmadığı	kategorik	halkalı=t, yok=f
F13	halka tipi	kategorik	örümcek ağı=c, geçici=e, genişleyen=r, oluklu=g, büyük=l, sarkık=p, kılıflı=s, zonlu=z, pullu=y, hareketli=m, yok=f
F14	habitat	kategorik	çimen=g, yaprak=l, çayır=m, patika=p, fundalık=h, kentsel=u, atık=w, orman=d
F15	mevsim	kategorik	ilkbahar=s, yaz=u, sonbahar=a, kış=w
C	sınıf	kategorik	yenilebilir=e, zehirli=p

Tablo 2. Veri setine ait birkaç örnek veri.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	C
5.48	x	s	n	t	s	d	r	4.47	14.14	w	f	f	d	a	p
4.84	c	i	y	f	a	c	n	6.99	8.08	w	f	f	d	u	p
6.68	x	t	n	f	s	c	n	6.52	8.76	n	f	f	d	w	p
55.82	o	y	y	f	p	c	y	8.92	48.05	k	f	f	d	s	e
46.9	o	y	y	f	p	c	y	6.12	39.68	k	f	f	d	s	e
5.11	b	y	r	f	s	c	n	6.84	9.73	n	t	e	d	u	p
8.19	b	t	w	f	e	c	w	13.2	14.82	w	t	l	d	a	p
8.34	x	t	n	f	x	c	w	7.4	21.38	w	f	f	d	a	e
3.16	f	s	w	f	d	d	w	3.33	5.73	w	f	f	g	a	e
3.37	f	t	u	f	a	c	w	4.26	7.05	w	f	f	d	a	p

4 Metot

4.1 Kodlama teknikleri

Bu çalışmada, KNN sınıflandırma algoritması için veri setindeki kategorik özellikleri sayısal özelliklere dönüştürmek amacıyla çeşitli kodlama teknikleri uygulanmıştır. Tablo 1'de görüldüğü üzere, veri setindeki 15 özelliğin 12 tanesinin kategorik formattadır. Bu özelliklerin etkili bir şekilde dönüştürülmesi sınıflandırma modelin performansını artırmak için çok önemlidir. Bu çalışmada kullanılan kodlama yöntemleri olan aşağıdaki açıklanmıştır.

4.1.1 Etiket kodlama

Etiket kodlama, kategorik değerleri tamsayı değerlerine dönüştürür. Her bir benzersiz kategori değere, alfabetik sıralamaya dayalı olarak benzersiz bir tamsayı atanır [19]. Örneğin, bir özelliğin kategorileri 'kırmızı', 'mavi', 'yeşil' ise, etiket kodlama bunları sırasıyla 0, 1, 2 olarak dönüştürebilir. Bu yöntem basit ve etkilidir, bu nedenle hesaplama açısından ucuz ve uygulanması kolaydır. Ancak, en büyük dezavantajı kategorik değişkenlere sıralama eklemesidir, bu da orijinal kategorik özellikte var olmayan bir sıralama veya derecelendirme anlamına gelir. Bu durum, belirli makine öğrenimi algoritmalarının kodlanmış değerleri sayısal bir öneme sahipmiş gibi yorumlamasına neden olabilir.

4.1.2 One-Hot kodlama

One-hot kodlama, kategorik özellikleri bir dizi ikili özelliğe dönüştürür. Bir özelliğin her kategorisi, yeni bir ikili sütuna (0 veya 1) dönüştürülür [20]. Örneğin, bir özelliğin kategorileri 'kırmızı', 'mavi', 'yeşil' ise, one-hot kodlama üç yeni ikili özellik oluşturur. 'Kırmızı' [1, 0, 0], 'mavi' [0, 1, 0] ve 'yeşil' [0, 0, 1] olarak temsil edilir. Bu yöntem, etiket kodlamanın sıralama sorununu ortadan kaldırsa da, özellikle birçok kategoriye sahip özellikler için boyutluluğun önemli ölçüde artmasına neden olabilir. Bu durum, özellikle bazı algoritmaların performansını olumsuz etkileyebilir. Ancak, kodlanmış özelliklerin özellik alanında eşit uzaklıkta olmasını sağlar, bu da KNN gibi mesafeye dayalı algoritmalar için faydalı olabilir.

4.1.3 Frekans kodlama

Frekans kodlama, her kategoriyi veri setindeki görülme sıklığıyla değiştirir. Bu yöntem, kategorik özelliğin dağılımını yakalar, bu da verinin içsel yapısını temsil etmede yararlı olabilir. Örneğin, renk özelliğinin kategorileri 'kırmızı', 'mavi', 'yeşil' ve normalize edilmiş sıklıkları 0.5, 0.3, 0.2 ise, 'kırmızı' 0.5, 'mavi' 0.3 ve 'yeşil' 0.2 olarak kodlanır [21].

4.1.4 Hash kodlama

Hash kodlama, kategorileri sayısal değerlere dönüştürmek için bir hash fonksiyonu kullanır. Bu yöntem, boyutluluğu azaltarak hem hesaplama açısından verimlilik sağlar hem de önemliliğe sahip özelliklerinin iyi bir şekilde ele alınmasını sağlar. One-hot kodlamasında olduğu gibi her kategori için yeni bir ikili sütun oluşturmak yerine, hash kodlama bir hash fonksiyonu kullanarak kategorileri belirli sayıda uzunluğa eşler. Bu durum, yüksek boyutluluk riskini azaltır ancak farklı kategorilerin aynı değere eşlendiği çakışmalar oluşturabilir [22].

4.1.5 Hedef kodlama

Hedef kodlama, bir kategoriyi o kategori için hedef değişkenin ortalaması ile değiştirir [23]. Bu yöntem, kategorik özellikler ile hedef değişken arasındaki ilişkiyi yakalar ve bu da modelin tahmin gücünü artırabilir. Örneğin, 'kırmızı' 0.7, 'mavi' 0.2 ve 'yeşil' 0.5 hedef ortalamasına karşılık geliyorsa, 'kırmızı' 0.7, 'mavi' 0.2 ve 'yeşil' 0.5 olarak kodlanır. Bu kodlama yöntemi doğrudan hedef bilgilerini içereceği için model performansını artırabilir, ancak küçük veri setlerinde aşırı uyuma (overfitting) yol açabilir. Aşırı uyumu azaltmak için düzenleme teknikleri ve çapraz doğrulama sıklıkla kullanılır.

4.2 K-En Yakın Komşu Algoritması (KNN: K-Nearest Neighbours)

K-En Yakın Komşu (KNN) algoritması, sınıflandırma ve regresyon görevleri için kullanılan basit ama güçlü bir gözetimli makine öğrenimi algoritmasıdır [3]. Bu çalışmada, parametrik olmayan yapısı ve kolay uygulanabilirliği nedeniyle sınıflandırma için KNN algoritması kullanılmıştır.

KNN, bir veri noktasını, k en yakın komşusunun çoğunluk sınıfına göre sınıflandırır. Başka bir ifadeyle, algoritma, en yakın örneklerden k örnek arar. Yeni örneğin hangi sınıfa düşeceğine karar verilir. Bu yeni örnek, k en yakın komşuda en çok oyu alan sınıfa ait olarak sınıflandırılır.

KNN algoritmasında veri noktalarında arasında uzaklığın hesaplanma biçimi, sınıflandırma performansını doğrudan etkileyen önemli bir faktördür. Bu uzaklık, düz çizgi mesafesi (diğer adıyla Öklid mesafesi), Manhattan mesafesi ve Chebyshev mesafesi olmak üzere üç farklı şekilde hesaplanabilir. Düz çizgi mesafesi, genellikle en sık tercih

edilen mesafe ölçütüdür ve iki nokta arasındaki doğrusal mesafeyi hesaplar. Bunun yanı sıra, Manhattan mesafesi, dik açılarla hesaplanan mutlak farklara dayanır ve özellikle verilerin grid yapısına sahip olduğu durumlarda etkilidir. Chebyshev mesafesi ise iki nokta arasındaki en büyük mutlak farkı alarak mesafeyi hesaplar ve genellikle maksimum mesafe bazlı ölçümlerde tercih edilir. Bu çalışmada, düz çizgi mesafesinin tercih edilmesinin sebebi, verilerin doğrusal bir dağılım sergilemesi ve bu ölçümün doğrusal veri yapıları için daha etkili sonuçlar sağlamasıdır. [24]. Her test örneği $x_i = \{x_1, x_2, x_3, \dots, x_n\}$ ile eğitim veri noktaları $e_i = \{e_1, e_2, e_3, \dots, e_n\}$ arasındaki düz çizgi mesafesi, Denklem (1) ile hesaplanabilir.

$$d(x,e) = \sqrt{\sum_{i=1}^{n}(x_i - e_i)^2} \tag{1}$$

En yakın komşuları bulma işlemi için, mesafe metriği dikkate alınarak x ile en yakın k adet eğitim örneği belirlenir. Bu komşular $N_k(x)$ olarak adlandırılır.

K değerinin seçimi KNN modelinin performansı için çok önemlidir. Küçük bir k değeri, modelin gürültüye duyarlı olmasına neden olabilirken, büyük bir k değeri karar sınırını yumuşatabilir. Bu çalışmada, optimal k değeri grid arama yöntemi ile belirlenmiştir.

Grid arama, bir makine öğrenimi modelinin hiper parametrelerini optimize etmek için kullanılan sistematik bir arama yöntemidir. Bu yöntem, hiper parametrelerin her biri için önceden belirlenmiş bir dizi değer oluşturur ve tüm olası kombinasyonları denenerek modelin performansı değerlendirilir. Bu yöntem, her kombinasyon için modeli eğitip bir değerlendirme üzerinden sonuçları karşılaştırır ve en iyi performans sağlayan parametre kombinasyonunu seçer. Bu yöntem ile elde edilen sonuçlar Bölüm 4'te paylaşılmıştır.

Son adım ise çoğunluk oylaması işlemi içerir. Yeni veri noktası x_i için sınıf etiketi $C(x_i)$, k adet en yakın komşusu arasında çoğunluk oylaması ile belirlenir. Bu işlem, Denklem (2) ile gerçekleştirilebilir.

$$C(x_i) = argmax_k \sum_{x_j \in N_k(x)} C(e_j, Y_k) \tag{2}$$

burada Y_k potansiyel sınıfı temsil ederken, C ise sınıf gösterge fonksiyonu olup, argümanın Y_k sınıfına ait olup olmadığını gösterir.

5 Bulgular ve tartışma

Bu çalışmada, KNN modelinin sınıflandırma performansını ölçmek için doruluk, kesinlik, duyarlılık ve f1-skoru olmak üzere dört farklı metrik kullanılmıştır [25, 26].

Doğruluk, modelin doğru sınıflandırdığı örneklerin toplam örneklere oranını ifade eder. Genel performansı ölçer ve Denklem (3) ile hesaplanır.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

burada, TP (True Positive) doğru pozitifleri, TN (True Negative) doğru negatifleri, FP (False Positive) yanlış pozitifleri ve FN (False Negative) yanlış negatifleri temsil eder.

Kesinlik, modelin pozitif tahminlerinin doğruluğunu ölçer. Bir sınıflandırma modeli pozitif bir tahmin yaptığında, bu tahminin ne kadarının doğru olduğunu ifade eder. Yüksek kesinlik, modelin pozitif tahminlerinde az hata yaptığını gösterir ve Denklem (4) ile hesaplanır.

$$Kesinlik = \frac{TP}{TP + FP} \tag{4}$$

Duyarlılık, modelin gerçek pozitif örnekleri doğru bir şekilde tanıma yeteneğini ölçer. Gerçek pozitiflerin ne kadarının doğru bir şekilde tahmin edildiğini ifade eder. Yüksek duyarlılık, modelin gerçek pozitifleri kaçırmadan doğru bir şekilde tahmin ettiğini gösterir ve Denklem (5) ile hesaplanır.

$$Duyarlılık = \frac{TP}{TP + FN} \tag{5}$$

F1-skoru, kesinlik ve duyarlılığı birleştirerek modelin genel performansını değerlendiren bir metriktir. Kesinlik ve duyarlılık arasındaki dengeyi sağlar ve dengesiz sınıf dağılımlarında özellikle faydalıdır. F1-skoru, Denklem (6) ile hesaplanır.

$$F1 - Skoru = 2 \cdot \frac{Kesinlik \cdot Duyarlılık}{Kesinlik + Duyarlılık} \tag{6}$$

KNN modelinin performansını daha güvenilir ve genel bir şekilde değerlendirmek için 5 katlı çapraz doğrulama kullanılmıştır. 5 katlı çapraz doğrulama, veri setini beş eşit parçaya böler ve her bir parça sırasıyla test seti olarak kullanılırken, kalan dört parça eğitim seti olarak kullanılır. Bu süreç beş kez tekrarlanır ve her bir katmanın sonuçları ortalama alınarak nihai performans metriği elde edilir. Bu yöntem, modelin genelleme yeteneğini daha iyi değerlendirmeye yardımcı olur ve aşırı uyum (overfitting) riskini azaltır [27]. Ayrıca, test seti boyutu %20 olarak belirlenmiştir, bu da veri setinin %80'inin eğitim için, %20'sinin ise modelin performansını bağımsız olarak değerlendirmek için kullanıldığı anlamına gelir.

Bu çalışmada yapılan analizler, Python programlama dili kullanılarak gerçekleştirilmiştir. Analizler, 6 çekirdekli ve 3.10 GHz hızında çalışan Intel(R) Core(TM) i5-10500 işlemciye sahip bir bilgisayarda yürütülmüştür. Ayrıca, 16 GB RAM kapasitesi ve 64-bit Windows 11 Pro işletim sistemi ile çalışılmıştır. Bu donanım ve yazılım kombinasyonu, büyük veri kümelerinin işlenmesi ve algoritmaların etkin bir şekilde uygulanması için yeterli performansı sağlamıştır.

Tablo 3, KNN modelinin farklı k değerleri ve kodlama teknikleri kullanılarak elde edilen performans sonuçlarını

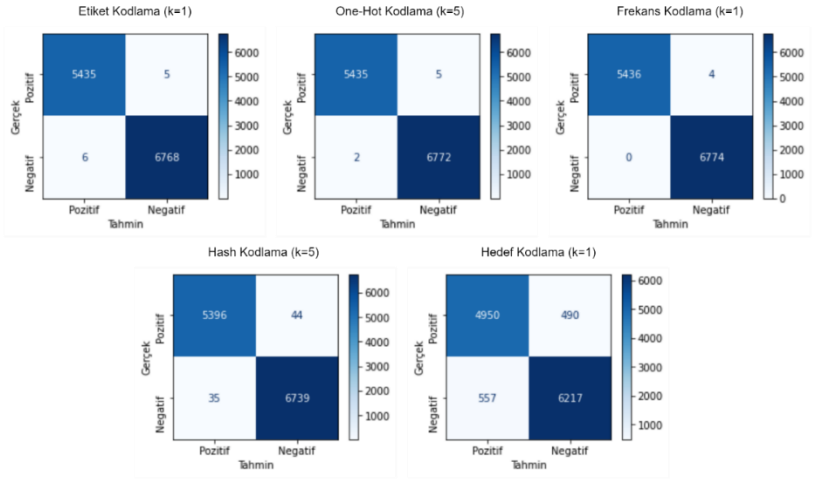
göstermektedir. Açıkça görüldüğü üzere, frekans kodlama k değerinin 1 olduğu durumda, 0.9994 kesinlik, 1.0 duyarlılık, 0.9997 f1-skoru ve 0.9997 doğruluk değerleri ile en yüksek performansı göstermiştir. En kötü performansı ise k değerinin 7 olduğu durumda, 0.9243 kesinlik, 0.9083 duyarlılık, 0.9162 f1-skoru ve 0.9079 doğruluk değerleri ile hedef kodlama göstermiştir. Ayrıca, etiket kodlama (0.9993 kesinlik, 0.9991 duyarlılık, 0.9992 f1-skoru ve 0.9991 doğruluk), frekans kodlama (0.9994 kesinlik, 1.0 duyarlılık, 0.9997 f1-skoru ve 0.9997 doğruluk değerleri ile) ve hedef kodlama (0.9269 kesinlik, 0.9178 duyarlılık, 0.9223 f1-skoru ve 0.9143 doğruluk değerleri ile) en iyi performanslarını k değerinin 1 olduğu durumda göstermişlerdir. One-hot kodlama (0.9993 kesinlik, 0.9997 duyarlılık, 0.9995 f1-skoru ve 0.9994 doğruluk değerleri ile) ve hash kodlama (0.9935 kesinlik, 0.9948 duyarlılık, 0.9942 f1-skoru ve 0.9935 doğruluk değerleri ile) ise en iyi performanslarını k değerinin 5 olduğu durumda elde etmişlerdir. Her bir kodlamanın en iyi performansları gösterdiği durumların karışıklık matrisleri (confusion matrix) Şekil 2'de verilmiştir.

Şekil 3, KNN modelinin farklı k değerlerine karşı kodlama çeşitlerinin doğruluklarının nasıl değiştiğini göstermektedir. Ayrıca, kesikli çizgili alanın büyütülmüş halini de sunmaktadır. Bu bölüm, özellikle yüksek doğruluk oranlarının küçük değişimlerini daha net görmek için büyütülmüştür.

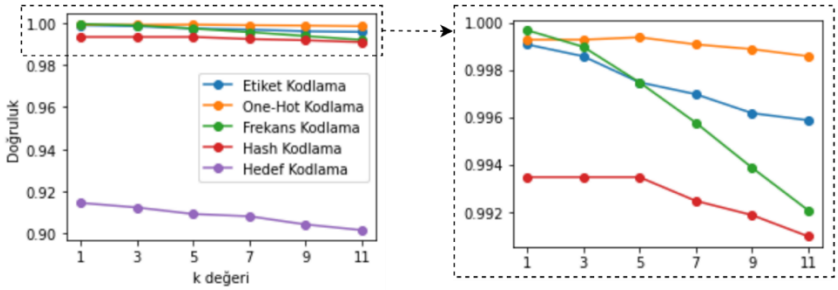
- Etiket kodlama, farklı k değerlerinde oldukça istikrarlı bir performans sergilemektedir. Doğruluk oranı neredeyse sabit kalarak 1.0'a çok yakın seyretmektedir.
- One-hot kodlama, düşük k değerlerinde yüksek doğruluk oranına sahip olup, k değeri arttıkça performansında hafif bir düşüş gözlemlenmektedir. Ancak doğruluk oranı genel olarak yüksek kalmaktadır.
- Frekans kodlama da yüksek doğruluk oranına sahiptir ve k değerinin artmasıyla birlikte performansında çok hafif bir düşüş görülmektedir.
- Hash kodlama, diğer kodlama tekniklerine kıyasla daha düşük doğruluk oranına sahiptir ve k değeri arttıkça performansında belirgin bir düşüş yaşanmaktadır.
- Hedef kodlama, diğer tekniklere göre en düşük doğruluk oranına sahip olup, k değeri arttıkça performansında sürekli bir düşüş yaşamaktadır.

Tablo 3. KNN modelinin farklı k değerleri ve kodlama teknikleri için performans sonuçları.

k değeri	Kodlama Tekniği	Kesinlik	Duyarlılık	F1-Skoru	Doğruluk
	Etiket Kodlama	0.9993	0.9991	0.9992	0.9991
	One-Hot Kodlama	0.9993	0.9994	0.9993	0.9993
1	Frekans Kodlama	**0.9994**	**1.0**	**0.9997**	**0.9997**
	Hash Kodlama	0.9926	0.9956	0.9941	0.9935
	Hedef Kodlama	0.9269	0.9178	0.9223	0.9143
	Etiket Kodlama	0.9987	0.9988	0.9987	0.9986
	One-Hot Kodlama	0.9991	0.9996	0.9993	0.9993
3	Frekans Kodlama	0.9982	**1.0**	0.9991	0.9990
	Hash Kodlama	0.9934	0.9948	0.9941	0.9935
	Hedef Kodlama	0.9246	0.9163	0.9204	0.9121
	Etiket Kodlama	0.9979	0.9975	0.9977	0.9975
	One-Hot Kodlama	0.9993	0.9997	0.9995	0.9994
5	Frekans Kodlama	0.9966	0.9990	0.9978	0.9975
	Hash Kodlama	0.9935	0.9948	0.9942	0.9935
	Hedef Kodlama	0.9227	0.9125	0.9175	0.9090
	Etiket Kodlama	0.9973	0.9972	0.9973	0.9970
	One-Hot Kodlama	0.9988	0.9996	0.9992	0.9991
7	Frekans Kodlama	0.9948	0.9976	0.9962	0.9958
	Hash Kodlama	0.9929	0.9937	0.9933	0.9925
	Hedef Kodlama	0.9243	0.9083	0.9162	0.9079
	Etiket Kodlama	0.9970	0.9960	0.9965	0.9962
	One-Hot Kodlama	0.9982	0.9997	0.9990	0.9989
9	Frekans Kodlama	0.9935	0.9954	0.9945	0.9939
	Hash Kodlama	0.9917	0.9937	0.9927	0.9919
	Hedef Kodlama	0.9197	0.9061	0.9128	0.9040
	Etiket Kodlama	0.9972	0.9954	0.9963	0.9959
	One-Hot Kodlama	0.9981	0.9994	0.9987	0.9986
11	Frekans Kodlama	0.9916	0.9941	0.9928	0.9921
	Hash Kodlama	0.9909	0.9929	0.9919	0.9910
	Hedef Kodlama	0.9177	0.9032	0.9103	0.9013



Şekil 2. Her bir kodlamanın en iyi performansları gösterdiği durumların karışıklık matrisleri



Şekil 3. Her bir kodlamanın KNN modelinin farklı k değerine karşı doğruluklarının değişimi

6 Sonuçlar

Bu çalışma, farklı kodlama tekniklerinin KNN algoritmasının mantarların sınıflandırma performansı üzerindeki etkilerini kapsamlı bir şekilde değerlendirmiştir. Bu sınıflandırma işlemi mantarların yenilebilir veya zehirli olduğunu tespit etmeye yaramaktadır. Sonuçlar, kodlama tekniklerinin model performansını önemli ölçüde etkilediğini göstermektedir. Özellikle frekans kodlama, düşük k değerlerinde en yüksek performansı sağlarken, hedef kodlama düşük performans sergilemiştir. Bu bulgular, araştırmacıların ve uygulayıcıların, veri setlerinin özelliklerine ve uygulama alanlarına bağlı olarak en uygun kodlama tekniklerini seçmelerine yardımcı olabilir. Gelecekteki çalışmalarda, farklı veri setleri ve makine öğrenimi algoritmaları üzerinde benzer analizler yaparak, kodlama tekniklerinin genelleştirilebilirliği daha da araştırabilir. Ayrıca, öznitelik seçimi ve boyut azaltma işlemleri yapılarak modelin performansı arttırılabilir. Ayrıca kullanılan modelin performansı ROC eğrisi ile desteklenebilir. Son olarak, bu çalışmanın bulguları, makine öğrenimi modellemelerinde kategorik veri dönüşümünün stratejik bir bileşen olarak ele alınması gerektiğini göstermektedir. Bu bağlamda, veri bilimi topluluğu, kodlama teknikleri ve veri önişleme yöntemleri üzerine daha fazla araştırma ve deney yaparak, bu alandaki bilgi birikimini ve uygulama pratiğini zenginleştirebilir.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %8

Kaynaklar

[1] C. Pan, A. Poddar, R. Mukherjee, and A.K. Ray, Impact of categorical and numerical features in ensemble machine learning frameworks for heart disease prediction. Biomedical Signal Processing and Control, 76, 103666, 2022. https://doi.org/10.1016/j.bspc.2022.103666.

[2] K.S. Sree, J. Karthik, C. Niharika, P.V.V.S. Srinivas, N. Ravinder, and C. Prasad, Optimized conversion of categorical and numerical features in machine learning models. In 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 294-299, IEEE, November 2021. https://doi.org/10.1109/I-SMAC52330.2021.9640967.

[3] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, KNN model-based approach in classification. in On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, pp. 986-996, November 3-7, 2003.

[4] H. Gupta and V. Asha, Impact of encoding of high cardinality categorical data to solve prediction

problems. Journal of Computational and Theoretical Nanoscience, vol. 17, no. 9-10, pp. 4197-4201, 2020. https://doi.org/10.1166/jctn.2020.9044.

[5] P. Yan, Anomaly Detection in Categorical Data with Interpretable Machine Learning: A random forest approach to classify imbalanced data. 2019.

[6] K. Budholiya, S.K. Shrivastava, and V. Sharma, An optimized XGBoost based diagnostic system for effective prediction of heart disease. Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 7, pp. 4514-4523, 2022. https://doi.org/10.1016/j.jksuci.2020.10.013.

[7] T. Al-Shehari and R.A. Alsowail, An insider data leakage detection using one-hot encoding, synthetic minority oversampling, and machine learning techniques. Entropy, vol. 23, no. 10, p. 1258, 2021. https://doi.org/10.3390/e23101258.

[8] M. Hosni, Encoding Techniques for Handling Categorical Data in Machine Learning-Based Software Development Effort Estimation. in KDIR, pp. 460-467, 2023.

[9] M.X. Low, T.T.V. Yap, W.K. Soo, H. Ng, V.T. Goh, J.J. Chin, and T.Y. Kuek, Comparison of label encoding and evidence counting for malware classification. Journal of System and Management Sciences, vol. 12, no. 6, pp. 17-30, 2022. https://doi.org/10.33168/JSMS.2022.0602.

[10] S.K. Das and M.Z. Rahman, A Study on Machine Learning Algorithms with Different Encoding Techniques for Identifying the Right One for Patients' Big Data. Jahangirnagar University Journal of Science, vol. 43, no. 1, pp. 63-78, 2021.

[11] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl, Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. Computational Statistics, vol. 37, no. 5, pp. 2671-2692, 2022. https://doi.org/10.1007/s00180-022-01207-6.

[12] A.S. Mohanty, K.C. Patra, and P. Parida, Toddler ASD Classification Using Machine Learning Techniques. International Journal of Online & Biomedical Engineering, vol. 17, no. 7, 2021. https://doi.org/10.3991/ijoe.v17i07.23497.

[13] S. Zhang, Y. Yuan, Z. Yao, X. Wang, and Z. Lei, Improvement of the performance of models for predicting coronary artery disease based on XGBoost algorithm and feature processing technology. Electronics, vol. 11, no. 3, p. 315, 2022. https://doi.org/10.3390/electronics11030315.

[14] L.B. Nascimento, M. de Sousa Balbino, M.L. Teodoro, and C.N. Nobre, Assessment of the Relationship Between Attribute Coding and the Interpretability of Machine Learning Models: An Analysis in the Context of Children and Adolescents with Depression. In BIOSTEC (2), pp. 482-489, 2024.

[15] F. Pargent, B. Bischl, and J. Thomas, A benchmark experiment on how to encode categorical features in predictive modeling. München: Ludwig-Maximilians-Universität München, 2019.

[16] D. Wagner, D. Heider, and G. Hattab, Mushroom data creation, curation, and simulation to support classification tasks. Scientific Reports, vol. 11, no. 1, p. 8134, 2021. https://doi.org/10.1038/s41598-021-87602-3.

[17] UCI Machine Learning Repository, Secondary Mushroom. https://archive.ics.uci.edu/dataset/848/secondary +mushroom+dataset, Accessed 25 June 2024.

[18] UCI Machine Learning Repository, Mushroom. https://archive.ics.uci.edu/dataset/73/mushroom, Accessed 25 June 2024.

[19] M.K. Dahouda and I. Joe, A deep-learned embedding technique for categorical features encoding. IEEE Access, vol. 9, pp. 114381-114391, 2021. https://doi.org/10.1109/ACCESS.2021.3104357.

[20] C. Seger, An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. 2018.

[21] C.T.T. Thuy, K.A. Tran, and C.N. Giap, Optimize the combination of categorical variable encoding and deep learning technique for the problem of prediction of Vietnamese student academic performance. International Journal of Advanced Computer Science and Applications, vol. 11, no. 11, 2020. https://doi.org/10.14569/IJACSA.2020.0111135.

[22] I. Lopez-Arevalo, E. Aldana-Bobadilla, A. Molina-Villegas, H. Galeana-Zapién, V. Muñiz-Sanchez, and S. Gausin-Valle, A memory-efficient encoding method for processing mixed-type data on machine learning. Entropy, vol. 22, no. 12, p. 1391, 2020. https://doi.org/10.3390/e22121391.

[23] S. Mumtaz and M. Giese, Hierarchy-based semantic embeddings for single-valued & multi-valued categorical variables. Journal of Intelligent Information Systems, vol. 58, no. 3, pp. 613-640, 2022. https://doi.org/10.1007/s10844-021-00693-2.

[24] A. Almomany, W.R. Ayyad, and A. Jarrah, Optimized implementation of an improved KNN classification algorithm using Intel FPGA platform: Covid-19 case study. Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 6, pp. 3815-3827, 2022. https://doi.org/10.1016/j.jksuci.2022.04.006.

[25] J. Lever, Classification evaluation: It is important to understand both what a classification metric expresses and what it hides. Nature Methods, vol. 13, no. 8, pp. 603-605, 2016.

[26] Ž. Vujović, Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 599-606, 2021.

[27] H. Jabbar and R.Z. Khan, Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). Computer Science, Communication and Instrumentation Devices, vol. 70, no. 10.3850, pp. 978-981, 2015.