



NECMETTİN ERBAKAN
ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Yapay Zekaya Giriş Dersi Proje Formu

Proje Başlığı	
KNN Algoritmasının Web Sitesi Üzerinde Dinamik Olarak Kullanılması	

Öğrenci Bilgileri	
Öğrenci No/Ad Soyad	20010011033 Erdem DEMİRCİ 21100011014 Erdem LALE 21100011010 Metin Furkan YAMAN

Danışman Unvan: Doç. Dr. Mehmet HACİBEYOĞLU
--

İçindekiler

1. Proje Hakkında	3
2. KNN Nedir?	4
2.1 K Değeri Seçimi.....	5
2.2 En Yakın Komşu Bulma Yöntemi.....	7
2.3 KNN Algoritması Karşılaştırma Metrikleri	8
2.4 KNN Algoritmasının Güçlü Yönleri	8
2.5 KNN Algoritmasının Zayıf Yönleri	9
2.6 KNN Algoritmasının Kullanım Alanları [1,7]	9
3. Projede Kullanılacak Donanım ve Yazılımlar	9
3.1.Donanım:	9
3.2. Yazılımlar ve Kütüphaneler:.....	10
4. KNN Algoritmasının Web Sitesi Üzerinde Kullanımı	11
4. Kaynakça	13

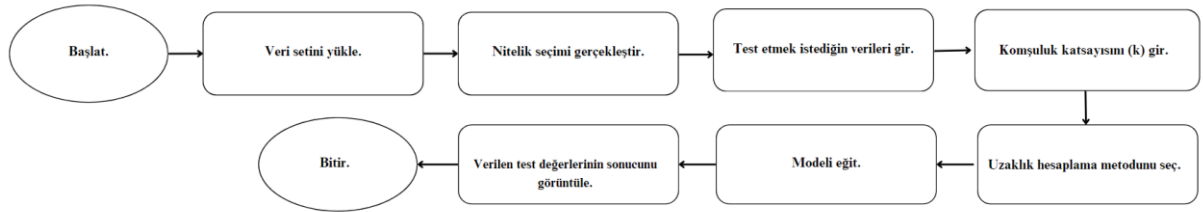
1. Proje Hakkında

Bu projede gerçekleştirilecek web uygulaması, kullanıcıların kendi veri setlerini yükleyip K-En Yakın Komşu (KNN) algoritmasını çalıştırabilecekleri bir platform sunmaktadır. Makine öğrenimi alanında pratik deneyim kazanmak isteyen kullanıcıların ihtiyaçlarını karşılayan bu uygulama, KNN algoritmasını kullanarak gerçek dünya problemlerini çözmeye yönelik bir ortam sağlamaktadır.

Projenin temel amacı, kullanıcıların KNN algoritmasını pratik bir şekilde deneyimleyebilmesidir. Kullanıcılar, kendi nümerik değer içeren veri setlerini kullanarak çeşitli senaryolara uygulayabilecekleri KNN algoritmasını test edebilmesini sağlar.

Kullanıcılar, kendi metriklerini ("k" komşu sayısı, uzaklık hesaplama metodu) belirleyerek algoritmayı özelleştirebilir ve farklı senaryolar için uygun olan en iyi parametreleri deneyebilirler. Bu özellik, kullanıcıların algoritmanın işleyişini daha iyi anlamalarını ve gerçek dünya problemlerine daha uygun çözümler üretmelerini sağlar.

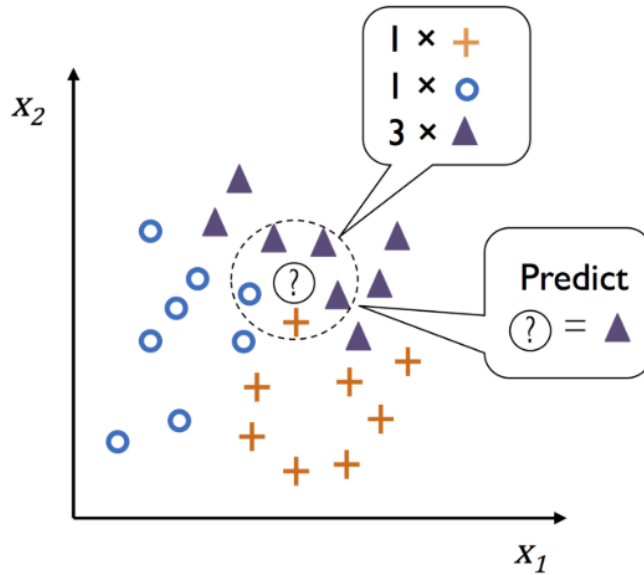
Sonuç olarak, bu interaktif web uygulaması, makine öğrenimi kavramlarını deneyimlemeyi kolaylaştırmak ve kullanıcıların KNN algoritmasını deneyimleyebilmesi amacıyla geliştirilmiştir. Kullanıcılar, kendi veri setlerini (Yalnızca ".csv" dosyalarına izin verilmiştir.) kullanarak gerçek dünya problemlerine pratik çözümler üretebilir ve bu sayede makine öğrenimi alanındaki yeteneklerini geliştirebilirler. Şekil 1'de web sitesinin çalışma mekanizmasını anlatan akış şeması gösterilmiştir.



Şekil 1. Web Sitesinin Çalışma Mekanizması

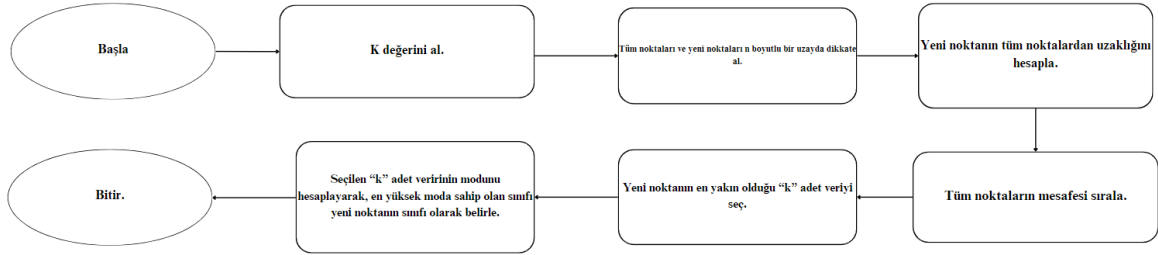
2. KNN Nedir?

K-Nearest Neighbors (KNN) algoritması, bireysel bir veri noktasının gruplandırılması hakkında sınıflandırmalar veya tahminler yapmak için yakınlığı kullanan, parametrik olmayan, denetimli bir öğrenme sınıflandırıcısıdır. KNN algoritması hem regresyon hem de sınıflandırma problemleri için kullanılabilir de genellikle benzer noktaların birbirine yakın bulunabileceği varsayımıyla çalışan bir sınıflandırma algoritması olarak kullanılır [1]. Sınıflandırma bağlamında, KNN modelinin en basit biçimi, belirli bir sorgu noktası için en benzer k eğitim örneği arasında en sık temsil edilen sınıf etiketini tahmin etmektir. Başka bir deyişle, sınıf etiketi, k eğitim etiketinin "modu" olarak düşünülebilir veya "çoğunluk oyu" ile belirlenebilir [2]. Bu yaklaşım, algoritmanın farklı kalıplara uyum sağlamasına ve verilerin yerel yapısına göre tahminlerde bulunmasına olanak tanır [3]. Çoğunluk oyu terimi, genellikle belirli bir veri noktası çevresinde en sık temsil edilen sınıf etiketinin kullanılmasını ifade eder. Örneğin, KNN algoritması bir veri noktasını sınıflandırırken, çevresindeki en yakın K komşudan elde edilen etiketler arasında çoğunluk oyu prensibini kullanarak yeni veri noktasına bir sınıf etiketi atar. Regresyon problemleri, sınıflandırma problemine benzer bir kavram kullanır, ancak bu durumda, bir sınıflandırma hakkında tahminde bulunmak için en yakın k komşunun ortalaması alınır. Buradaki temel ayrım, sınıflandırmanın ayrık değerler için kullanılması, regresyonun ise sürekli değerler için kullanılmasıdır. Sürekli hedefin hesaplanması için yaygın bir yaklaşım, k en yakın komşunun üzerindeki hedef değerinin ortalamasını veya ortalama değerini hesaplamaktır [1].



Şekil 2. K=5 olan bir 3-sınıf problemi için KNN'in görselleştirilmesi [2]

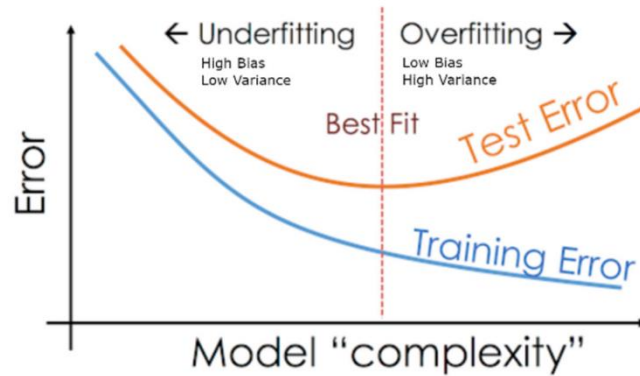
Özetle, K-En Yakın Komşu (KNN) algoritmasında ilk olarak k değeri belirlenir ve gelen verinin diğer verilere olan mesafesi hesaplanır. Komşular arasındaki mesafeyi hesaplarken veri setindeki bazı özellikler baskın olabilmektedir. Baskın olan değerlerin normalize edilmesi önemlidir. Normalizasyon, veri setindeki özelliklerin farklı ölçeklerde veya dağılımlarda olması durumunda, bu özelliklerin benzer bir ölçek veya dağılıma dönüştürülmesi işlemidir. Hesaplamalar sonucunda, sınıflandırma için algoritma en yakın ' k ' komşunun çoğunluk oyunu seçer. Regresyon için ise algoritma genellikle en yakın ' k ' komşunun değerlerinin ortalamasını alır ve bu ortalamaya dayanarak tahmin edilen değeri belirler. Şekil 2'de KNN ile sınıflandırma algoritmasının akış şeması verilmiştir.



Şekil 2. KNN Sınıflandırma Algoritması Akış Şeması

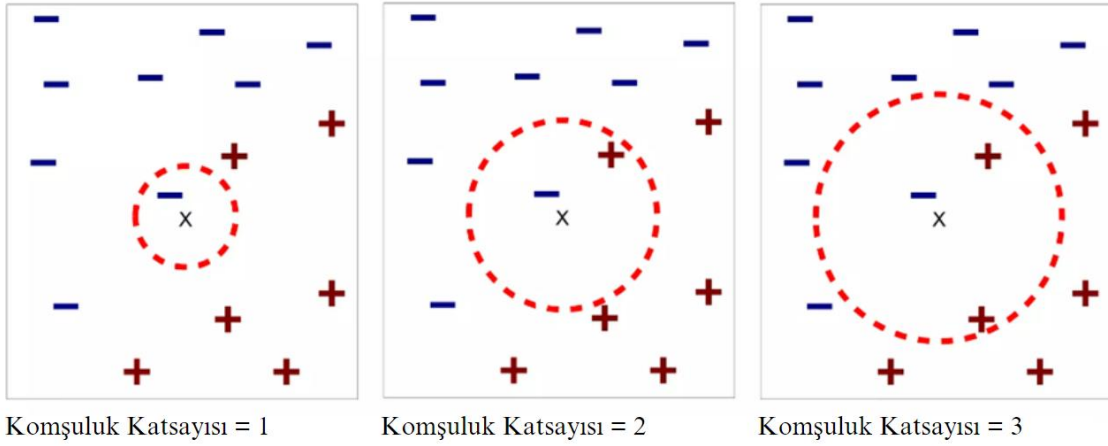
2.1 K Değeri Seçimi

KNN algoritmasında kullanılan k değeri, belirli bir sorgu noktasının sınıflandırılmasında dikkate alınacak komşu sayısını tanımlar. Örneğin, eğer $k=1$ ise, sorgu noktası tek en yakın komşusunun sınıfına atanır. k değeri modelin overfitting ya da underfitting olmasına sebebiyet verebilir. Düşük k değerleri genellikle yüksek varyansla ilişkilendirilirken, düşük önyargıya (bias) sahip olabilir; buna karşılık, daha yüksek k değerleri genellikle yüksek önyargıya ve düşük varyansa neden olabilir [1].



Şekil 3. Underfitting, Overfitting Gösterimi [6]

Büyük veri kümelerinde, gürültülü verilerin varlığı tahmin işlemlerinin doğruluğunu olumsuz etkileyebilir. Bu sorunun çözümünde kritik bir rol oynayan unsurlardan biri, uygun bir k değeri seçimidir. K değerinin doğru bir şekilde belirlenmesi, en yakın k komşuluklarının belirlenmesi ve sonrasında yapılacak olan oylama işleminin güvenilirliğini sağlar. Gürültülü verilerin etkisini azaltmak için daha yüksek bir k değeri seçilebilir. Ayrıca, k değeri tek sayılar olarak seçilerek, oylama işlemi sonucunda eşit oy alınması durumundan kaçınılmaya çalışılır. Bu yaklaşım, tahmin doğruluğunu artırmaya ve gürültülü verilerin etkisini minimize etmeye yönelik etkili bir stratejidir.



Şekil 4. Komşuluk Katsayısının Çalışma Mantığı [5]

2.2 En Yakın Komşu Bulma Yöntemi

KNN algoritmasında en yakın komşuları bulmak için çeşitli yöntemler mevcuttur. Bu yöntemler şunlardır:

- **Öklid Uzaklığı:** İki nokta arasındaki kartezyen mesafe uzaklığıdır.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (1)$$

- **Manhattan Mesafesi:** N boyutlu bir vektör uzayındaki iki nokta arasındaki mesafe ölçüsüdür. Karşılık gelen boyutlardaki koordinatlar arasındaki mutlak mesafenin toplamı olarak tanımlanır [4].

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

- **Minkowski Mesafesi:** Bu mesafe ölçüsü, Öklid ve Manhattan mesafe ölçümlerinin genelleştirilmiş şeklidir. Aşağıdaki formülde yer alan p parametresi diğer mesafe metriklerinin oluşturulmasına olanak sağlar. Manhattan mesafesi, p=1 olduğunda bu formülle temsil edilir ve Öklid mesafesi, p=2 olduğunda gösterilir [1].

$$d(x, y) = \left(\sum_{i=1}^n (x_i - y_i)^p \right)^{1/p} \quad (3)$$

2.3 KNN Algoritması Karşılaştırma Metrikleri

KNN algoritmasının başarısını ve performansını değerlendirebilmek için çeşitli karşılaştırma metrikleri kullanılır. Bu metrikler, modelin doğruluğunu, hassasiyetini, özgüllüğünü ve diğer performans ölçütlerini belirlemek için kullanılır. KNN algoritması, özellikle parametre seçimlerine ve veri setinin yapısına oldukça duyarlıdır, bu nedenle karşılaştırma metriklerinin dikkatlice değerlendirilmesi büyük önem taşır.

Tablo 1. kullanılan metrikler ve formülasyonları		
Metrik	Formülasyon	
Doğruluk (Accuracy): Modelin ne kadar doğru çalıştığı belirlenir. Doğru tahminlerin toplam veri kümesine oranı ile hesaplanır. Doğruluk skoru 0 ve 1 arasında olup 1'e yaklaşan skorlarda model başarılı kabul edilir.	$\frac{TP + TN}{TP + TN + FP + FN}$	(4)
Kesinlik (Precision): Doğru Pozitif olarak tahmin edilen verilerin ne kadarının pozitif olduğunu gösterir. Doğru pozitif (TP) tahminlerin ile tüm pozitif (TP + FP) tahmin veri kümesine oranı ile hesaplanır.	$\frac{TP}{TP + FP}$	(5)
Duyarlılık (Recall): Modelin pozitif vakaları tahmin etme kabiliyetidir. Doğru pozitif örneklerin ne kadarının doğru bir şekilde pozitif olarak tahmin edildiğini gösterir. Doğru pozitif tahminin (TP) doğru pozitif ile yanlış negatif değerlerin toplamına oranlanarak hesaplanır.	$\frac{TP}{TP + FN}$	(6)
F1-skoru: Kesinlik ve duyarlılık arasındaki harmonik ortalamadır. 0 ve 1 arasında olup 1'e yaklaşan skorlarda model başarılı kabul edilir.	$\frac{2 * Precision * Recall}{Precision + Recall}$	(7)
TP (True Positive): Gerçek değeri pozitif olan bir verinin, model tarafından da pozitif olarak tahmin edilmesi. FP (False Positive): Gerçek değeri negatif olan bir verinin, model tarafından pozitif olarak tahmin edilmesi. TN (True Negative): Gerçek değeri negatif olan bir verinin, model tarafından da negatif olarak tahmin edilmesi. FN (False Negative): Gerçek değeri pozitif olan bir verinin, model tarafından negatif olarak tahmin edilmesi.		

2.4 KNN Algoritmasının Güçlü Yönleri

- Diğer makine öğrenimi algoritmalarına kıyasla, temel KNN algoritmasının nispeten az hiper parametresi vardır.
- Nispeten kolay uygulanabilir ve daha kolay yorumlanabilir.
- Yeni eğitim örnekleri eklendikçe, tüm eğitim verileri hafızada saklandığından algoritma yeni verileri hesaba katacak şekilde kolayca ayarlanır.
- Gürültüye sahip veriler için sağlamdır.

2.5 KNN Algoritmasının Zayıf Yönleri

- KNN tembel bir algoritma olduğundan diğer sınıflandırıcılara göre daha fazla bellek ve veri depolama alanı kaplar. Bu hem zaman hem de para açısından maliyetli olabilir. Daha fazla bellek ve depolama, işletme giderlerini artıracak ve daha fazla verinin hesaplanması daha uzun sürecektir.
- Yeni bir örneği sınıflandırmak (bir örneğin diğer örneklerle mesafesinin hesaplanması ve karşılaştırılması) çok zaman alır.
- KNN algoritması boyutluluk lanetinin (curse of dimensionality) kurbanı olma eğilimindedir, bu da yüksek boyutlu veri girişlerinde iyi performans göstermediği anlamına gelir. Algoritma en uygun özellik sayısına ulaştıktan sonra, ek özelliklerin sayısı sınıflandırma hatalarının miktarını artırır [1].

2.6 KNN Algoritmasının Kullanım Alanları [1,7]

1. İstatiksel Sınıflandırma (Hastalık tespiti, Yağış tespiti vb.)
2. DNA Dizilim Analizi
3. Öneri Sistemleri
4. Pazar Analizi
5. Görüntü İşleme

3. Projede Kullanılacak Donanım ve Yazılımlar

3.1.Donanım:

- İşlemci: 11. Nesil i7-11800H
- Ekran Kartı: NVIDIA RTX 3050Ti
- Depolama: 512 GB SSD
- RAM: 16 GB

3.2. Yazılımlar ve Kütüphaneler:


- **Python:** Python, yorumlanan, nesne yönelimli, yüksek seviyeli bir programlama dilidir ve dinamik semantiğe sahiptir. Yüksek seviyeli yerleşik veri yapıları ve dinamik bağlama gibi özelliklere sahiptir. Python veri analizi ve makine öğrenmesi için geniş bir kütüphaneye sahiptir [8]. Bu sayede makine öğrenmesi projelerinde kullanımı oldukça tercih edilen bir dildir.
- **JavaScript:** JavaScript, modern web geliştirme süreçlerinde kritik bir rol oynayan bir programlama dilidir. İlk olarak Netscape Communications Corporation tarafından geliştirilmiş olup, sonrasında ECMA International tarafından standartlaştırılmıştır [9]. JavaScript, istemci tarafında çalışır ve tarayıcı ortamında yürütülür. Bu dille yazılan kodlar, web sayfalarında dinamik içerik oluşturmak, kullanıcı etkileşimlerini yönetmek, form doğrulama işlemleri gerçekleştirmek ve sayfa stilini değiştirmek gibi çeşitli görevleri yerine getirebilir.
- **HTML:** HTML (Hyper Text Markup Language), web sayfalarının oluşturulmasında kullanılan standart bir işaretleme dilidir [10]. HTML, belirli bir web sayfasının yapısını tanımlamak için kullanılan etiketlerden oluşur. Bu etiketler, metin, resimler, bağlantılar ve diğer medya öğeleri gibi içerikleri belirtmek için kullanılır. HTML, tarayıcılar tarafından yorumlanarak kullanıcılara görsel ve interaktif web sayfaları sunar.
- **CSS:** CSS (Cascading Style Sheets), HTML öğelerinin ekranda görüntüleneceğini tanımlayan bir stil dilidir. CSS, bir web sitesinin görünümünü kontrol etmek için kullanılır ve HTML ile kullanıldığında sayfalara stil ve düzen kazandırır [11].
- **Django:** Django, Python dilinde yazılmış, açık kaynaklı, yüksek seviyeli bir web çerçevesidir. Hızlı, güvenli ve ölçeklenebilir web uygulamaları geliştirmeyi kolaylaştırmak için tasarlanmıştır [12]. MVC (Model-View-Controller) tasarım desenine dayanır, ancak Django'nun kendi kuralları ve yapıları vardır. Django, veritabanı işlemleri, oturum yönetimi, URL yönlendirmesi, HTML şablonları, form işlemleri, güvenlik ve yetkilendirme gibi birçok web geliştirme görevini kolaylaştıran birçok yerleşik özellik sunar.
- **Numpy:** NumPy, Python'da bilimsel hesaplama için temel pakettir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler ve diziler üzerinde hızlı işlemler için çeşitli rutinler sağlayan bir Python kütüphanesidir. Bu rutinler arasında matematiksel, mantıksal, şekil manipülasyonu, sıralama, seçme, kesikli Fourier dönüşümleri, temel doğrusal cebir, temel istatistiksel işlemler, rastgele simülasyonlar ve çok daha fazlası bulunur [13].
- **Pandas:** Pandas, Python'da gerçek dünya veri analizi için temel bir yapı taşı olarak tasarlanmıştır. Aynı zamanda, herhangi bir dilde mevcut olan en güçlü ve esnek açık kaynaklı veri analizi ve manipülasyon aracı olma hedefine sahiptir. Pandas, veri kümeleri üzerinde bölme, uygulama, birleştirme gibi işlemleri gerçekleştirmek için güçlü bir grupta motoruna sahiptir [14].
- **Scikit-learn:** Python programlama dili için açık kaynaklı bir makine öğrenimi kütüphanesidir. Temel olarak, sınıflandırma, regresyon, kümeleme, boyut azaltma, model seçimi ve model değerlendirme gibi çeşitli makine öğrenimi algoritmalarını ve araçlarını içerir [15].

4. KNN Algoritmasının Web Sitesi Üzerinde Kullanımı

Geliştirdiğimiz uygulama, kullanıcı dostu arayüzü ve basit 4 adımdan oluşan kullanımı ile dikkat çekmektedir. Kullanıcılar, bu adımları takip ederek KNN algoritmasını kolaylıkla uygulayabilir ve verilerini analiz edebilirler.

1. Adım

Kullanıcı, projeye başlamak için CSV Dosyası Yükle butonuna tıklayarak analiz etmek istediği dosyayı seçer ve yükler. Bu adımda "Dosya Seç" butonuna tıklanarak yüklemek istenen dosya (örneğin, Iris.csv) bilgisayardan seçilir. Dosya seçildikten sonra "Yükle" butonuna basılarak dosya sisteme yüklenir. Bu adım şekil 5'te detaylı olarak gösterilmiştir.



The screenshot shows a web interface for a K-NN project. At the top, it says 'Yapay Zeka dersi K-NN projesi'. Below that, the title 'CSV Dosyası Yükle' is displayed. There is a button labeled 'Dosya Seç' next to a text input field that contains 'Iris.csv'. Below the input field is a green button labeled 'Yükle'.

Şekil 5. Veri Seti Yükleme Ekranı

2. Adım

Kullanıcı, veri kümesinin niteliklerini ve sınıfını seçer. Bu adımda, analizde kullanılacak özellikler (nitelikler) ve hedef sınıf belirlenir. Kullanıcı, KNN algoritması için minimum ve maksimum k değerlerini girer. Şekil 6'daki örnekte minimum k değeri 1, maksimum k değeri ise 15 olarak belirlenmiştir. Buradaki amaç kullanıcının verdiği sınırlar arasındaki en iyi "k" değerini GridSearchCV yardımıyla bulmaktır. Son olarak, "Gönder" butonuna tıklanarak seçilen özellikler ve sınıf ile K değerleri sisteme gönderilir. "Veri Sayfası" butonu, kullanıcının veri setini değiştirebilmesini sağlar.

Niteliklerinizi Seçiniz

Id int64

SepalLengthCm float64

SepalWidthCm float64

PetalLengthCm float64

PetalWidthCm float64

Species object

Sınıfı Seçiniz

Id int64

SepalLengthCm float64

SepalWidthCm float64

PetalLengthCm float64

PetalWidthCm float64

Species object

Minimum K değerini giriniz

1

Maximum K değerini giriniz

15

Gönder

Veri Sayfası

Şekil 6. Nitelik ve Sınıf Seçim Ekranı

3. Adım

Kullanıcı, tahmin etmek istediği örneğin değerlerini girer. Aynı zamanda KNN algoritmasının parametreleri olan K değeri ve uzaklık hesaplama metriğini seçerek KNN algoritmasını dener. Ayrıca, GridSearch Algoritmasına göre en iyi uzaklık metriği ve KNN komşu değeri kullanıcıya önerilir. Bu kısım, Şekil 7'de detaylı olarak gösterilmiştir.

Tahmin etmek istediğiniz örneğin değerlerini giriniz.

SepalLengthCm

6.5

SepalWidthCm

3.2

PetalLengthCm

5.1

PetalWidthCm

2.0

K Değeri

5

Öklid

Manhattan

Minkowski - Öklid

Minkowski - Manhattan

Sonuç

Veri Sayfası

GridSearch Algoritmasına göre;

Tavsiye edilen k değeri: 5

Tavsiye edilen metric yöntemi: euclidean

Şekil 7. Test Verisi Ekleme ve KNN Parametre Seçimi Ekranı

4. Adım

Kullanıcı, modelin performansını değerlendirir. Bu adımda, modelin eğitim ve test doğruluğu ile sınıf değerlerine göre precision, recall ve f1-score gibi değerlendirme metrikleri gösterilir. Kullanıcı, modelin genel doğruluğunu ve her bir sınıf için performans ölçümlerini analiz eder. Şekil 8'de bu adım detaylı olarak gösterilmiştir.

Sınıf Değeri	Eğitimin Doğruluğu	Testin Doğruluğu
Iris-virginica	0.9642857142857143	0.9736842105263158

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	13
Iris-versicolor	1.00	0.94	0.97	16
Iris-virginica	0.90	1.00	0.95	9
accuracy			0.97	38
macro avg	0.97	0.98	0.97	38
weighted avg	0.98	0.97	0.97	38

Şekil 8. Model Sonuçlarını Görüntüleme Ekranı

4. Kaynakça

[1] IBM. “KNN”. <https://www.ibm.com/topics/knn>

Erişim tarihi: 21 Mart 2024

[2] Raschka, S. (2018). STAT 479: Machine Learning Lecture Notes içinde (s.7). Department of Statistics, University of Wisconsin–Madison.

[3] Geeksforgeeks. “K-nearest neighbours”. <https://www.geeksforgeeks.org/k-nearest-neighbours/> Erişim tarihi: 21 Mart 2024

[4] Geeksforgeeks. <https://www.geeksforgeeks.org/how-to-calculate-manhattan-distance-in-r/> Erişim tarihi: 21 Mart 2024

[5] Gunawardena, T. (2016). K-En Yakın Komşu (KNN) Algoritması Konu Anlatımı. Algorithms K- Nearest Neighbors

[6] Guerrero, Jose & Coltelli, Mauro & Marsella, Maria & Celauro, Angela & Antonio, Jose. (2022). Convolutional Neural Network Algorithms for Semantic Segmentation of Volcanic Ash Plumes Using Visible Camera Imagery. Remote Sensing. 14. 4477. 10.3390/rs14184477.

[7] Sun, J., Du, W., & Shi, N. (2018). A survey of kNN algorithm. Information Engineering and Applied Computing, 1(1), 1-10.

[8] Python. "About". <https://www.python.org/about/>

Erişim tarihi: 21 Mart 2024

[9] Mozilla Developer Network (MDN). "JavaScript." <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Erişim tarihi: 22 Mart 2024.

[10] W3Schools. "HTML Introduction". https://www.w3schools.com/html/html_intro.asp.

Erişim tarihi: 22 Mart 2024.

[11] W3Schools. "CSS Introduction". https://www.w3schools.com/css/css_intro.asp.

Erişim tarihi: 22 Mart 2024.

[13] Numpy. "What Is Numpy" <https://numpy.org/doc/stable/user/whatisnumpy.html>

Erişim tarihi: 22 Mart 2024

[14] Pandas. "About Pandas". <https://pandas.pydata.org/about/index.html>

Erişim tarihi: 22 Mart 2024

[15] VanderPlas, J. (2017, s.343). *Machine Learning with Scikit-Learn: A Hands-On Guide*. O'Reilly Media.