FUTURE STEM HUB

Uygulamalı Python ve Scratch Eğitim Materyalleri

Uygulamalı Python ve Scratch Görevleri Eğitim Materyalleri

FUTURE-STEM-HUB

Yapay Zeka Eğitimi ile Lise Öğrencileri ve Öğretmenleri için STEM Eğitimi Güçlendirme Projesi

Proje No.
2024-1-DE03-KA220-SCH-000247346

Co-funded by the European Union

UYGULAMALI PYTHON VE SCRATCH EĞITIM MATERYALLERI

“Yapay Zeka Eğitimi ile Lise Öğrencileri ve Öğretmenleri için STEM Eğitimi Güçlendirme Projesi / FUTURE-STEM-HUB” projesi (referans no: 2024-1-DE03-KA220-SCH-000247346), Avrupa Birliği Erasmus+ Programı tarafından ortak finanse edilmektedir. Proje, Duisburg-Essen Üniversitesi (Almanya) tarafından koordine edilmekte olup, dört ortak kuruluşu daha içermektedir: M&M Profuture Training (İspanya), Kütahya İl Milli Eğitim Müdürlüğü (Türkiye), COOPETAPE - Eğitim Kurumu, ETAP Okulu'nun denetleme kuruluşu CRL (Portekiz) ve Tetra Solutions Ltd. (Bulgaristan).

Proje ekibi üyeleri, tüm ortak kuruluşları temsil ederek, her bir ortağın bir modülün yazarlığından sorumlu olduğu, uygulamalı Python ve Scratch görevleri içeren FUTURE-STEM-HUB Eğitim Materyallerini geliştirdiler. Amaçları, ortaöğretim öğrencilerine yapay zekanın temel kavramlarını tanıtmak ve toplumsal ve etik etkileri konusunda farkındalık ve tartışma ortamı yaratmaktır..

Authors:
Mustafa Bilgin, University of Duisburg-Essen (Almanya)
Monica Moreno, M&M Profuture Training (İspanya)
Montserrat Renedo, M&M Profuture Training (İspanya)
João Barroso, ETAP School (Portekiz)
Angelina Presa, ETAP School (Portekiz)
Silviya Georgieva, Tetra Solutions Ltd. (Bulgaristan)
Borislava Zaharieva-Tomova, Tetra Solutions Ltd. (Bulgaristan)
Yeliz Yurter, Kütahya MEM (Türkiye)
Özcan Turan, Kütahya MEM (Türkiye)

Editor:
Mustafa Bilgin, University of Duisburg-Essen

Şekiller Listesi

Düzen Şablonu: Tetra Solutions Ltd.
Kapak Resmi: Hibrit oluşturma (Konsept & çizim taslağı: Mustafa Bilgin,
dijital rötuş: Google Gemini kullanılarak oluşturulmuştur).
S. 6: Tetra Solutions Ltd. tarafından hazırlanan grafik.
S. 7, 8: Hibrit oluşturma (Konsept & çizim taslağı: Mustafa Bilgin,
dijital rötuş: Google Gemini kullanılarak oluşturulmuştur).
Simgeler Flaticon.com (Yüksek kaliteli simgeler) adresinden alınmıştır.

Sorumluluk Reddi ve Güvenlik

Bu materyallerde sağlanan programlama kodlarının kullanımı ve harici web sitelerine veya üçüncü taraf platformlara erişim, tamamen kullanıcıların kendi sorumluluğundadır. Proje ekibi; Google Colab, Scratch veya Machine Learning for Kids gibi harici bağlantıların içeriği, işlevselliği veya kullanılabilirliği konusunda hiçbir sorumluluk kabul etmez. Etkileşimli görevlerin çoğu aktif bir internet bağlantısı ve verilerin bulut sunucularında işlenmesini gerektirdiğinden, güvenlik standartlarına ve sistem gereksinimlerine (güncel tarayıcı sürümleri gibi) uyma sorumluluğu kullanıcılara aittir.

Yapay zeka modelleri oluşturmak için üçüncü taraf araçlar kullanılırken hiçbir kişisel veya hassas bilgi yüklenmemelidir. Eğitim materyalleri 15-18 yaş grubuna yönelik olduğundan, harici hizmetlere kayıt olma ve çevrimiçi etkinliklerin gerçekleştirilmesi, ilgili sağlayıcıların kullanım koşullarına uygun olarak ve ideal olarak bir öğretmenin veya yasal vasinin gözetimi ve onayı ile yapılmalıdır. Sağlanan kod örneklerinin çalıştırılmasından kaynaklanan herhangi bir veri kaybı veya kullanılan uç cihazlardaki bozulmalar için sorumluluk kabul edilmez.

Kısaltma Listesi:

(AI) : Yapay Zeka

(CNN): Evrişimsel Sinir Ağı

(CV): Bilgisayar Görüşü

(EACEA): Avrupa Eğitim ve Kültür Yürütme Ajansı

(FPS): Saniyede Kare Sayısı

IBM: Uluslararası İş Makineleri

(IDE): Entegre Geliştirme Ortamı

(ML): Makine Öğrenimi

(ML4K) : Çocuklar için Makine Öğrenimi

(MIT) Massachusetts Teknoloji Enstitüsü

(MNIST): Ulusal Standartlar ve Teknoloji Enstitüsü (Veri Kümesi)

NN: Sinir Ağı

(NLP) : Doğal Dil İşleme

RGB : Kırmızı, Yeşil, Mavi (Renk modeli)

STEM. Bilim, Teknoloji, Mühendislik ve Matematik

TTS : Metinden Sese Dönüştürme

UNESCO: Birleşmiş Milletler Eğitim, Bilim ve Kültür Örgütü

YOLO: Sadece Bir Kez Bakarsın

İçerik

Empowering Secondary School STEM Education with AI Training and Resources for Students and Educators
2024-1-DE03-KA220-SCH-000247346

Project Overview

FUTURE-STEM-HUB

FUTURE-STEM-HUB projesi, yapay zekâ (YZ) konularının ortaokullardaki STEM eğitimine entegrasyonunu şu yollarla geliştirmeyi ve kolaylaştırmayı amaçlamaktadır: 1) YZ kavramlarını ve toplumsal etkilerini tanıtan eğitim materyalleri sağlamak; 2) Öğrencilerin Python programlama kullanarak YZ'yi keşfetmeleri için pratik öğrenme kaynakları sunmak ve 3) Öğretmenlere YZ'yi ortaokul STEM eğitimine entegre etmeleri için destek sağlamak..

Proje Çıktıları

1. **1. Ders: Dijital Giriş: Yapay Zeka Temelleri (Lise Öğrencileri için Etkileşimli Eğitim Materyalleri Aracılığıyla Yapay Zekaya Giriş)**

2. **2. Kurs: Python ve Scratch ile Yapay Zekaya Daha Derinlemesine Bakış (İleri Yapay Zeka: Lisel Öğrencileri için Uygulamalı Öğrenme Materyalleri)**

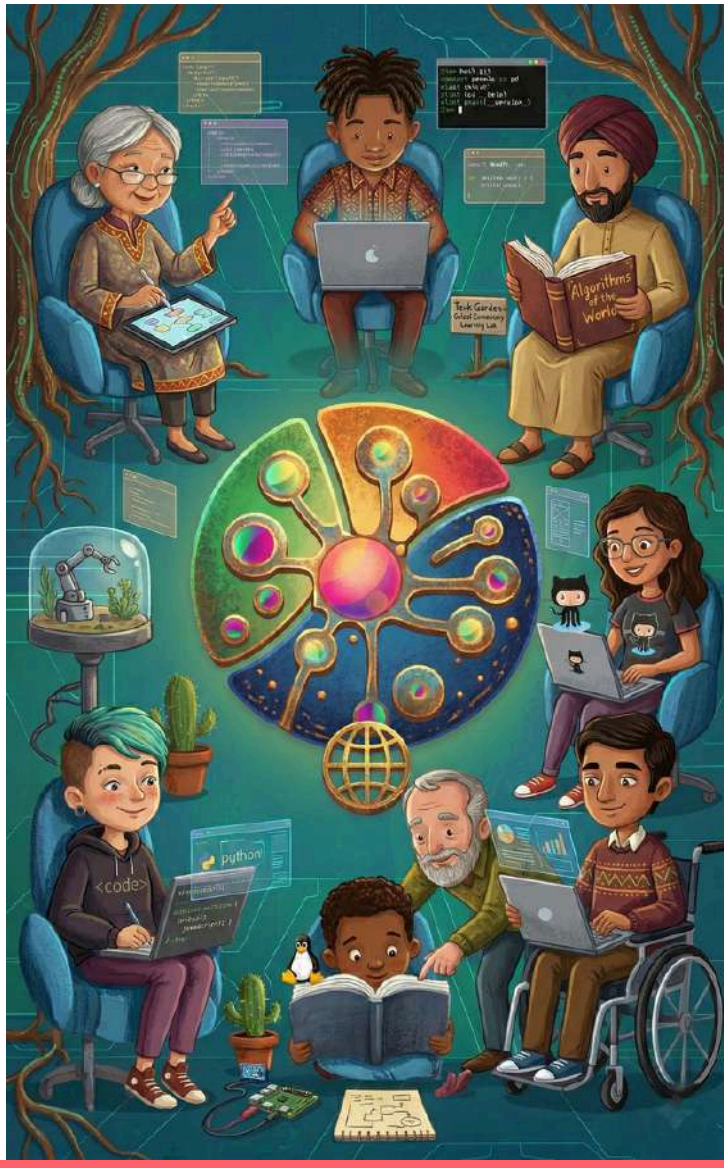
3. **Okul Eğitimcileri için E-Araç Seti: Yapay Zeka Becerilerini Geliştirme (Ortaöğretim Öğretmenleri için Yapay Zeka Metodolojik Kılavuzu)**

GİRİŞ

Python ve Scratch ile Uygulamalı Görevler İçeren Eğitim Materyallerine Hoş Geldiniz—lise öğrencilerini yapay zekanın büyüleyici ve pratik dünyasına tanıtmak için tasarlanmış, ilgi çekici ve etkileşimli bir öğrenme yolculuğu. Lise öğretmenlerini ve öğrencilerini, Python ve Scratch aracılığıyla temel yapay zeka kavramlarını keşfetmek için erişilebilir, uygulama odaklı materyallerle donatmak amacıyla hazırlanmıştır. Çeşitli STEM yeterlilik seviyelerine sahip 15-18 yaş arası öğrenciler için özel olarak tasarlanan materyaller, kendi hızlarında eş zamanlı olmayan öğrenme için mükemmeldir ve öğretmen rehberliğinde sınıf öğretimine sorunsuz bir şekilde entegre edilebilir.

İçerik, Python ve Scratch'te pratik programlama becerilerine, oyun tabanlı öğrenme (Kaçış Odası) yoluyla matematiksel mantığa, yaratıcı yapay zeka uygulamalarına, verilerle gerçek dünya problemlerini çözmeye ve etik hususlara odaklanan beş kapsamlı modüle ayrılmıştır.

Her modül, teoriyi bir araya getirir, uygulanmış testlere atıfta bulunur, daha fazla keşif için harici platformlar kullanır ve yapay zeka kavramları ve metodolojilerinin kapsamlı bir şekilde anlaşılmasını sağlamak için pratik alıştırmalar içerir. Beş modülün tamamı için tahmini öğrenme süresi yaklaşık 10 saattir.

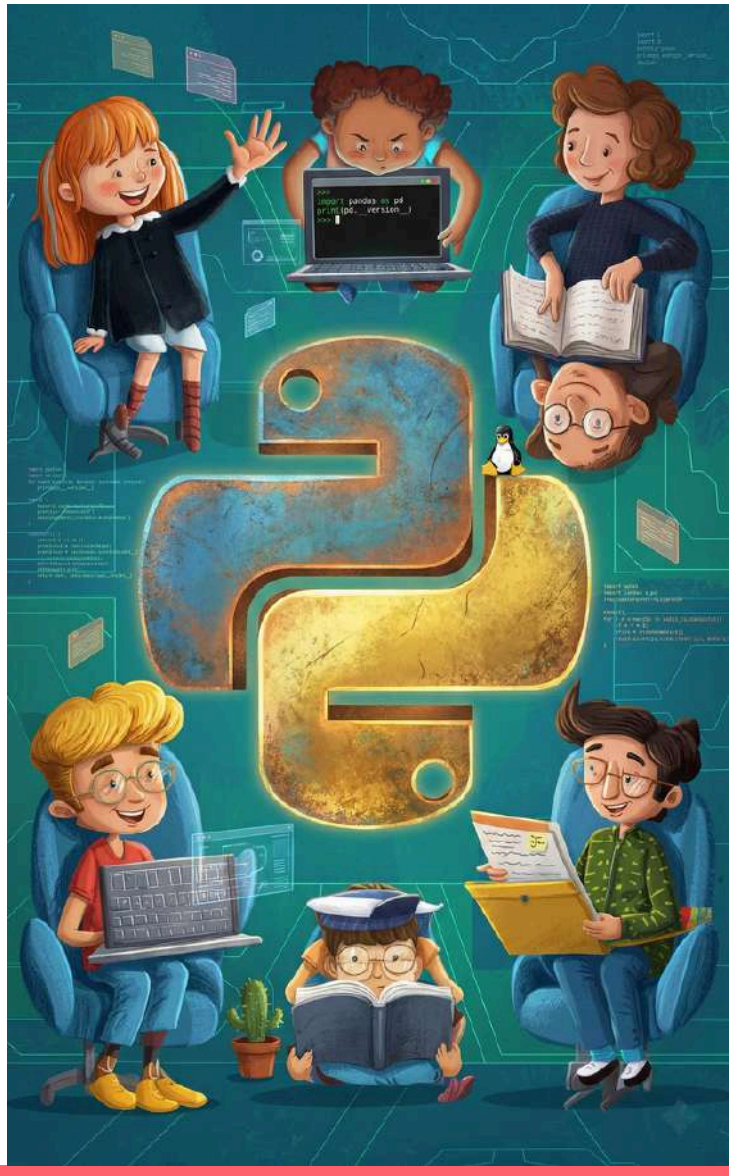


FUTURE-STEM-HUB

Kursu tamamladıktan sonra öğrenciler, makine öğrenimi, sinir ağları, derin öğrenme gibi temel yapay zeka kavramlarını anlayabilecek ve tanımlayabileceklerdir. Yapay zekanın tarihi ve evrimi ile çeşitli sektörlerdeki uygulamaları hakkında daha derin bilgi edinecek, yeni yapay zeka teknolojilerinin günümüz trendlerini ve dünyasını nasıl şekillendirdiğini öğreneceklerdir. Öğrenciler ayrıca bilgisayar görüşü ve doğal dil işleme gibi temel teknolojilere daha derinlemesine dalacaklardır. Son olarak, yapay zekanın etik hususları, yapay zekanın sorumluluğu ve toplumsal etkisi hakkında daha derin bir anlayış kazanacaklardır.

Öte yandan, öğretmenler, müfredatlarını ve öğretimlerini desteklemek için STEM sınıflarında kolayca kullanabilecekleri ve uygulayabilecekleri yenilikçi, etkileşimli kaynaklara erişebileceklerdir. Bu materyaller, ortaöğretim öğrencilerini öğrenme sürecine aktif olarak dahil etmek için farklı yaklaşımlar sunarak daha ilgi çekici ve etkileyici sınıf ortamları oluşturmalarına yardımcı olacaktır.

Bu materyaller daha sonra FUTURE STEM-HUB platformunda barındırılacak ve proje web sitesi olan www.future-stem-hub üzerinden erişilebilecek bir çevrimiçi kursa dönüştürülecektir.



Kursu başarıyla tamamlayan öğrenciler, başarılarını belgeleyen dijital bir sertifika alacaklardır.

Modül 1: Mini Kodlama Zorlukları: Makine Öğrenimi ve Sinir Ağları

GİRİŞ

Akıllı telefonunuzun hangi videoları beğenebileceğinizi nasıl bildiğini hiç merak ettiniz mi? Ya da bir uygulamanın el yazınızı nasıl tanıyabildiğini? Tüm bunların arkasında, birçok insanın düşündüğünden çok daha az gizemli olan Yapay Zeka (YZ) var! Bu bölümde, Aylin ve onun zeki akıllı saati Alara'ya katılacaksınız. Birlikte, YZ'nin gerçekte nasıl çalıştığını ortaya koyan maceralara atılacaksınız. Sinir Ağlarının (YS) ve Makine Öğrenmesinin (MAK) temel prensipleri ve bileşenleri hakkında net bir anlayış geliştireceksiniz; bunlar girdileri nasıl işlediklerini, verilerden nasıl öğrendiklerini ve nasıl kararlar aldıklarını içerecektir. En iyi yanı? Programlama uzmanı olmanıza gerek yok! Her macera, üç zorluk seviyesinde küçük kodlama zorlukları içerir. Başlangıç seviyesinden gerçek zorluğa kadar size en uygun olanı seçin. Göreceksiniz: daha ilk bölümden sonra, makinelerin nasıl "düşünmeyi öğrendiğini" anlayacaksınız. Gelin birlikte YZ dünyasına dalalım.

Modülün sonunda, aşağıdaki gibi farklı beceriler kazanabileceksiniz:

- **Yapay Zekaya Giriş:** Gizemli Hediye: Sadece Tıkırdamaktan Daha Fazlasını Yapan Bir Saat
- **Sinir Ağları — Temel Bilgiler:** Alara Uyanıyor
- **Karar Ağaçları:** Okul Güzergahı Dedektifi—Saatiniz Nasıl Düşünüyor?
- **Kural Tabanlı Sınıflandırma:** Spotify İsteklerinizi Nasıl Anlayabilir?
- **Pekiştirmeli Öğrenme:** Ödüller Yoluyla Öğrenme
- **Evrişimsel Sinir Ağları:** Yapay Zeka Görüntülerdeki Kenarları Nasıl Tespit Ediyor?

- **Doğal Dil İşleme:** Bilgisayarlar Dili Nasıl Anlıyor?
- **Öneri Sistemleri:** YouTube Ne İzlemek İstediğinizi Nasıl Biliyor?
- **Rastgele Ormanı:** Hava tahmini
- **Yapay Zeka Etiği:** Geleceğe Yönelik Vizyon—Yapay Zeka ile Neler Yapabiliriz ve Yapmalıyız

Modül Süre	**2 saat (1 saat öğrenme + 1 saat uygulamalı egzersizler)**

EĞİTİM MATERYALLERİ

AYLIN 17 yaşında, meraklı ve şeylerin nasıl çalıştığını çözmeyi çok seviyor. Asla pes etmiyor!

LENA, Aylin'in en iyi arkadaşıdır. Yaratıcıdır ve müziği sever. Ailesi Polonyalıdır; bazen lezzetli pierogi getirir!

ALARA, sıradan bir akıllı saat değil. O, Aylin'e yapay zekâ hakkında bilgi veren bir yapay zekâ.

Şekil 1. Eğitim karakterleri Aylin, Alara ve Lena; Konsept ve tasarım: Mustafa Bilgin; Google Gemini kullanılarak geliştirilmiştir.

ÜNİTE 1.1 DERS KONSEPTİ VE YAKLAŞIMI

Öğrenme süreci, STEM konularını öğrencilerin ilişki kurabileceği gerçek yaşam bağlamlarıyla birleştiren hikaye tabanlı bir çerçeve tarafından yönlendirilir. Sinir ağları veya pekiştirmeli öğrenme gibi karmaşık kavramlar, basitleştirilmiş, prototip biçiminde temsil edilerek öğrencilerin temel bir anlayış geliştirmelerine olanak tanır; önceden programlama deneyimi gerekmez. Bu materyal, bilgisayar bilimi ve teknik eğitim, STEM projeleri, ders dışı kulüpler veya disiplinler arası öğrenme ortamlarına mükemmel bir şekilde uyum sağlar. Sadece teknik anlayışı teşvik etmekle kalmaz, aynı zamanda Yapay Zeka Etiği bölümü aracılığıyla ahlaki ve sosyal sorular üzerine eleştirel düşünmeyi de teşvik eder.

ÜNİTE 1.2 YAPAY ZEKAYA (YZ) GİRİŞ

Etkinliklere dalmadan önce, neyle uğraştığımızı anlamak çok önemlidir. Yapay Zeka (AI), insan gibi düşünmek ve eylemlerini taklit etmek üzere programlanmış makinelerde insan zekasının simülasyonunu ifade eder. Bu terim, öğrenme ve problem çözme gibi insan zihniyle ilişkilendirilen özellikler sergileyen herhangi bir makine için de kullanılabilir.

Bu modülde, etkileşimli hikayeler aracılığıyla AI'nin temel yapı taşlarını keşfedeceğiz:

- **Sinir Ağları:** Bunlar, hayvan beyinlerini oluşturan biyolojik sinir ağlarından esinlenilmiş hesaplama sistemleridir. Yapay zekanın örneklerden "öğrenmesine" yardımcı olurlar.
- **Karar Ağaçları:** Kararları ve olası sonuçlarını ağaç benzeri bir modelle gösteren bir karar destek aracıdır. "Eğer bu olursa, o zaman şunu yap" mantığını temsil eder.
- **Makine Öğrenimi (ML):** Yapay zekanın bir alt kümesi olup, sistemlere açıkça programlanmadan deneyimlerden otomatik olarak öğrenme ve gelişme yeteneği kazandırır.

Bu modül, karmaşık matematiksel kavramları erişilebilir ve kolay anlaşılır hale getirmek için anlatısal bir yaklaşım (hikaye anlatımı) kullanmaktadır.

ÜNİTE 1.3 SİSTEM KURULUMU VE ÖN KOŞULLAR

Hangi platforma ihtiyacınız var?

Bu modüldeki Python kodunu çalıştırmak için Google Colab'ı kullanacaksınız. Google Colab, kodunuzu doğrudan web tarayıcınızda yazmanıza ve çalıştırmanıza olanak tanıyan ücretsiz bir çevrimiçi platformdur; kurulum gerekmez.
Buradan erişebilirsiniz:
https://colab.research.google.com/

Kayıt Olmanız Gerekiyor mu?

Evet, Google Colab'ı kullanmak için ücretsiz bir Google hesabına (Gmail hesabı) ihtiyacınız var. Zaten bir hesabınız varsa, doğrudan giriş yapabilirsiniz. Google hesabınız yoksa, başlamadan önce bir tane oluşturmalısınız.
Google hesabı oluşturmak için:
accounts.google.com adresine gidin ve kayıt adımlarını izleyin.

Kayıt İşlemine Neler Dahildir?

1. Google Colab'a gidin.
2. Sağ üst köşedeki "Giriş yap" seçeneğine tıklayın.
3. Google e-posta adresinizi ve şifrenizi girin.
4. Giriş yaptıktan sonra, mevcut bir not defterini açabilir veya yeni bir not defteri oluşturabilirsiniz.

Kodun Yazılması Beklenen Yer

1. Modüldeki tüm kodlama alıştırmaları bir Google Colab not defterinde hazırlanmıştır. Kod yazmanıza gerek yok; tüm alıştırmaları ve talimatları içeren önceden hazırlanmış bir not defteri kullanacaksınız.
İşte 1. Modülün not defteri:
>1. Modül Colab Not Defteri<

Defterle ne yapmalı:

1. Yukarıdaki bağlantıyı açın.
2. Kendi düzenlenebilir sürümünüzü kaydetmek için "Drive'a Kopyala" seçeneğine tıklayın.
3. Not defterinin içindeki talimatları izleyin ve her bir kod hücresini adım adım çalıştırın.

ÜNİTE 1.4 BAŞLAMADAN ÖNCE EK TALİMATLAR

1. Chrome, Firefox, Edge veya Safari gibi modern bir web tarayıcısı kullanın.
2. İnternet bağlantınızın stabil olduğundan emin olun; Google Colab çevrimiçi çalışır.
3. Kodlama görevine başlamadan önce her istasyon için hikayeyi ve talimatları dikkatlice okuyun.
4. Her istasyon üç zorluk seviyesi sunar:
   Temel seviye yeni başlayanlar içindir.
   İleri seviye, Python konusunda biraz deneyiminiz varsa uygundur.
   Uzman seviye: Gerçekten zorlu mücadeleler istiyorsanız, Uzman seviye tam size göre.
   Becerilerinize uygun seviyeyi seçin
5. Kod hücrelerini sırayla çalıştırın. Bazı alıştırmalar önceki alıştırmalara bağlıdır.
6. Takılıp kalırsanız, her alıştırmada verilen "İpucu" ve "Çözüm parçası"nı kullanın.
7. Bir görevi tamamladıktan sonra, sonucun nasıl etkilendiğini görmek için kodu değiştirmeyi deneyin.
   Deneme yapmak öğrenmenize yardımcı olur.

KAYNAKLAR

Emoji ve Simge Atıfı

- Bu belge, Unicode Konsorsiyumu tarafından tanımlanan standart Unicode emojilerini kullanmaktadır. Belirli emoji sürümleri, kullanıcının işletim sistemine bağlı olarak değişiklik gösterebilir. Unicode Konsorsiyumu. (2022). https://unicode.org/emoji/charts/full-emoji-list.html adresinden alınmıştır. Simgeler 'Flaticon.com'dan, Yüksek Kaliteli Simgeler'den alınmıştır.

Kitaplar ve Makaleler

Russell, S., & Norvig, P. (2021). Yapay Zeka: Modern Bir Yaklaşım (4. baskı). Pearson. (Sinir ağları, karar ağaçları ve pekiştirmeli öğrenme dahil olmak üzere yapay zeka kavramları üzerine kapsamlı bir ders kitabı.)

- Géron, A. (2022). Scikit-Learn, Keras ve TensorFlow ile Uygulamalı Makine Öğrenimi (3. baskı). O'Reilly. (Python ile makine öğrenimi ve sinir ağlarına pratik bir giriş.)

Yapay Zeka Etiği ve Toplum

Avrupa Komisyonu. (2019). Güvenilir Yapay Zeka için Etik Kılavuz İlkeleri. (Etik yapay zeka geliştirme ve dağıtımına yönelik bir çerçeve.)

UNESCO. (2021). Yapay Zeka Etiği Hakkında Tavsiye Kararı. (İnsan hakları ve kapsayıcılığa odaklanan küresel yapay zeka etiği kılavuzları.)

Python Programming for Beginners

- Sweigart, A. (2019). Automate the Boring Stuff with Python (2nd ed.). No Starch Press. (Beginner-friendly Python book with practical projects.)

LINKS TO LEARNING PLATFORMS

- Here is the notebook for Module 1 – https://colab.research.google.com/drive/14VaJnlgo5habZ0-N9jpz9nWZSVAw_Xwa?usp=sharing

- Python Official Documentation – https://docs.python.org/3/ (Official Python language reference and tutorials.)

PRATİK ALIŞTIRMA

İstasyon 1: Gizemli Hediye: Sadece Tıkırdamaktan Daha Fazlasını Yapan Bir Saat

Aylin'in 17. doğum günü—baklava ve balonların arasında gizemli küçük bir paket duruyor. Aylin paketi açmakta tereddüt ediyor.

AYLIN: "Bu nedir, Sema Teyze?"

SEMA TEYZE: "Özel bir şey canım. Büyükannen, 'Bu saatin bir ruhu var' derdi."

Aylin paketi dikkatlice açıyor. İçinde turkuaz mavisi kayışlı zarif bir akıllı saat var. Aniden ekran aydınlanıyor.

ALARA (saat): "Merhaba Aylin! Ben Alara—yeni yapay zekâ arkadaşın!"

Aylin şaşkınlıkla geriye sıçrıyor.

AYLIN: "Saat... konuşuyor mu?"

SEMA TEYZE: "Anladığın her dili konuşuyor. Alara sana yapay zekânın nasıl çalıştığını gösterecek!"

ALARA: “Beyniniz gibi olduğumu hayal edin: Öğrenen nöronlarım ve bağlantı kuran sinapslarım var.
Birlikte, makinelerin nasıl düşünmeyi öğrendiğini keşfedeceğiz!”

```
TEMEL
KOD:
NAME = INPUT("ADINIZ NEDIR?")
PRINT("MERHABA " + ____ + "! BEN ALARA, YAPAY ZEKÂ
ARKADAŞINIZ")
```

İpucu: Boşluğa girdiğiniz değişkeni kullanın.

İstasyon 1: Gizemli Hediye: Sadece Tıkırdamaktan Daha Fazlasını Yapan Bir Saat

GELİŞMİŞ

KOD:

```
PRINT("   AYLIN'IN DOĞUM GÜNÜ PLANLAMASI   ")
FAVORITE_FOOD = INPUT("EN SEVDIĞIN YEMEK NE?
(PIZZA/KEBAP/HAMBURGER): ")
DRINK = INPUT("VE YANINDA NE IÇMEK ISTERSIN? ")

IF FAVORITE_FOOD.LOWER() == "PIZZA":

PRINT("   ALARA DIYOR KI: PIZZA BENIM DE EN SEVDIĞIM!")
ELSE:

PRINT("   ALARA DIYOR KI: " + __ + " HARIKA OLUR!")

print("Ve yanında " + __ + "? Mükemmel kombinasyon!")
```

Çözüm parçası: favori_yemek, içecek

pucu: Boşlukları girdi değişkenlerinizle doldurun.

2. İSTASYON: Sinir Ağları — Alara Uyanıyor

Ertesi sabah Aylin parkta oturmuş Alara'yı izliyordu.

AYLIN: “Yani... benim gibi mi düşünüyorsun?”

ALARA: “Neredeyse! ‘Beynimi’ büyük bir düşünme fabrikası gibi hayal et. Öğrenen nöronlarım var. Bak...”

AYLIN: “Yani, benimle birlikte mi öğreneceksin?”

ALARA: “Evet,” diye yanıtladı Alara.

“Temelleri biliyorum, ama senin sayende dünyayı insan gözüyle görmeyi öğreniyorum.”

2. İSTASYON: Sinir Ağları — Alara Uyanıyor

TEMEL

KOD:

```
PRINT("YENI BIR OYUN ALMALI MIYIM?")
PRINT("0 ILE 1 ARASINDA PUAN VERIN, 0 = HIÇ DEĞIL, 0.5 =
TAMAM, 1 = ÇOK ISTIYORUM")

TASARRUF_BAKIYESI = FLOAT(INPUT("KUMBARANIZ NE KADAR DOLU?
(0-1): "))
ARZU = FLOAT(INPUT("OYUNU NE KADAR ISTIYORSUNUZ? (0-1): "))

DEF KARAR_NÖRONU(PARA, DILEK):

IF PARA * DILEK > ____:

RETURN "   EVET, AL!"

ELSE:

RETURN "   HAYIR, BEKLEMEK DAHA IYI"

print(karar_nöronu(tasarruf_bakiyesi, arzu))
```

İpucu: Çarpım 0,5'ten büyük olduğunda bir nöron 'ateşlenir'.

Çözüm parçası: 0,5

2. İSTASYON: Sinir Ağları — Alara Uyanıyor

GELİŞMİŞ (Bölüm I)

Kod:

```
print("İyi bir arkadaşlık mı?")
print("0 ile 1 arasında puan verin, 0 = hiç değil, 0.5 = iyi,
1 = çok iyi")
güven = float(input("Bu kişiye ne kadar güveniyorsunuz? (0-
1): "))
eğlence = float(input("Birlikte ne kadar eğleniyorsunuz? (0-
1): "))
yardımseverlik = float(input("Bu kişi ne kadar yardımsever?
(0-1): "))
# Ağırlıklar - arkadaşlıkta en önemli olan nedir?

Güven_ağırlığı = 0,5
Eğlence_ağırlığı = 0,3
Yardım_ağırlığı = 0,2

Arkadaşlık_analizi_tanımı(güven, eğlence, yardım):

girdiler = [güven, eğlence, yardım]

ağırlıklar = [güven_ağırlığı, eğlence_ağırlığı,
yardım_ağırlığı]
toplam = 0
girdiler için aralık(boyut(girdiler)):

toplam += girdiler[i] * ____
toplam döndür
```

2. İSTASYON: Sinir Ağları — Alara Uyanıyor

GELİŞMİŞ (Bölüm II)

Kod:

```
score = friendship_analysis(trust, fun, helpfulness)
print("Arkadaşlık Puanı:", score)
if score > 0.7:

print("   Gerçek arkadaşlık!")
elif score > 0.4:
print("   İyi bir tanıdık")
else:
print("   Belki de en iyi arkadaşlık değil")
```

Çözüm parçası: weights[i]

Kod: Her bir girdiyi karşılık gelen ağırlığıyla çarpın.

3. İSTASYON: Karar Ağaçları — Okul Güzergahı Dedektifi

Aylin otobüs durağında. Yağmur yağıyor. Alara ile okula nasıl gideceği hakkında konuşuyor.

ALARA: “Okul güzergahın için bir fikrim var! Hadi bir karar ağacı oluşturalım.”

Alara ona seçim yapmasında yardımcı oluyor – buna karar ağacı deniyor.

AYLIN: “Harika! Artık her zaman hangi yoldan gideceğimi bileceğim,” diyor Aylin mutlu bir şekilde.

ALARA: “Ve ben de senin nasıl karar verdiğini öğreniyorum.”

3. İSTASYON: Karar Ağaçları — Okul Güzergahı Dedektifi

TEMEL

Kod:

```
hava durumu = input("Hava nasıl? (güneşli/yağmurlu): ")

def okul_rotası(hava durumu):
if hava durumu == "yağmurlu":
return "   Şemsiyeyi al!"

else:
return "   ____"
print(okul_rotası(hava durumu))
```

İpucu: Yağmur yağmıyorsa ne yaparsınız?

Çözüm özeti: Şemsiyesiz dışarı çıkın!

GELİŞMİŞ

Kod:

```
hava durumu = input("Hava durumu? (yağmur/güneş): ")
otobüs_geliyor = input("Otobüs geliyor mu? (evet/hayır): ")

def plan_rota(hava durumu, otobüs_geliyor):
if hava durumu == "yağmur" ve otobüs_geliyor == "evet":
return "   Otobüse binin!"
elif hava durumu == "yağmur" ve otobüs_geliyor == "hayır":
return "   ____"

else:
return "   Yürüyün!"
print(plan_rota(hava durumu, otobüs_geliyor))
```

İpucu: Otobüs gelmezse, sadece şemsiye yardımcı olur.

Çözüm özeti: Şemsiyeyi alın!

4. İSTASYON: Kural Tabanlı Sınıflandırma — Spotify Nelerden Hoşlandığınızı Nasıl Bilebilir?

Aylin ve arkadaşı Lena müzik dinliyorlar.

LENA: “Saatin rock müzik sevdiğimi nereden biliyor?”

ALARA: “Bu kural tabanlı sınıflandırma! Benzer şarkıları gruplandırıyorum.”

LENA: “Bu sihir gibi!” diye haykırıyor Lena.

ALARA: “Sihir değil,” diye düzeltiyor Alara.
“Sadece kalıp tanıma—ve senin müzik zevkini her geçen gün daha iyi öğreniyorum.”

TEMEL

Kod:

```
song = input("Şarkınızı tanımlayın: ... (ör. gitar)")

def sort_music(song):
if "guitar" in song:
return "   Rock"
elif "____" in song:
return "   Pop"
else:
return "   Other"

print(sort_music(song))
```

İpucu: Dans edilebilir şarkılar genellikle pop müziktir.

Basitleştirilmiş Gruplandırma: Bu alıştırmada, dinleyicileri müzik tercihlerinin daha güçlü olduğu gruplara ayırıyoruz. K-Means gibi gerçek yapay zeka sistemleri, benzer zevklere sahip kullanıcıları otomatik olarak bulmak için daha karmaşık matematiksel yöntemler kullanır.

5. İSTASYON: Pekiştirmeli Öğrenme — Ödül Yoluyla Öğrenme

Aylin tabletinde bir köpek eğitim uygulamasıyla oynuyor.

AYLIN: "Bu sanal köpeğe ne yapması gerektiğini nasıl öğreteceğim?"

ALARA: "Çok basit: ödüllerle! Doğru bir şey yaptığında ona bir ödül ver.
Bu şekilde hangi eylemlerin yapmaya değer olduğunu yavaş yavaş öğrenir. Buna Pekiştirme Öğrenmesi denir!"

AYLIN: "Yani okul gibi—doğru cevaplar için iyi notlar mı?"

ALARA: "Aynen öyle! Ama burada 'doğru'nun ne olduğuna sen karar veriyorsun.
Köpek bir şeyler deniyor ve neyin işe yaradığını hatırlıyor."

AYLIN: "Yani sanal köpeğin gibi sen de öğreniyorsun?" diye soruyor Aylin gülümseyerek.
ALARA: "Bir bakıma evet," diye itiraf ediyor Alara. "Ödüller tüm öğrenenler için işe yarar."

TEMEL (Bölüm I)

Kod:

```
import random
print("Sanal köpeğinizi eğitin!")
print("Ödül olarak bir ikram verin (1) veya görmezden gelin
(0)\n")
actions = ["Otur", "Yat", "Pati Ver"]
probabilities = [0.3, 0.3, 0.4] # Başlangıç değerleri
disturb_chance = 0.2 # Köpeğin başka bir şey yapma olasılığı
%20
def normalize(probabilities):
total = sum(probabilities)
return [p / total for p in probabilities]
```

5. İSTASYON: Pekiştirmeli Öğrenme — Ödül Yoluyla Öğrenme

TEMEL (Bölüm II)

Kod:

```
for round in range(1, 6):
print(f"\n Eğitim seansı {tur}/5")

# Köpek tercihlerine göre bir eylem seçer
action = random.choices(actions, weights=_________________)[0]
# Boşluk 1

# Rahatsızlık: Köpek farklı bir eylem yapabilir
if random.random() < disturb_chance:
possible_actions = [a for a in actions if a != action]
wrong_action = random.choice(possible_actions)
print(f"Köpek '{action}' yapmalıydı ama bunun yerine şunu
yaptı:    {wrong_action}")
action = wrong_action
else:
print(f"Köpek şunu yaptı:    {action}")
# Kullanıcı etkileşimi ve giriş kontrolü
while True:
user_input = input("Ödül ver? (1=evet, 0=hayır): ").strip()

if user_input in ("1", "0"):
ödül = tamsayı(kullanıcı_girdisi)
break
print("Lütfen yalnızca 1 veya 0 girin!")

index = actions.index(action)")
```

İpucu: Otobüs gelmezse, sadece şemsiye yardımcı olur.

Çözüm özeti: Şemsiyeyi alın!

5. İSTASYON: Pekiştirmeli Öğrenme — Ödül Yoluyla Öğrenme

TEMEL (Bölüm III)

Kod:

```
# Öğrenme mantığı
Eğer ödül == 1 ise:
olasılıklar[indeks] += ______ # 2'lik boşluk
"   Köpek mutlu: 'Bu iyiydi!'")
aksi takdirde:

olasılıklar[indeks] = max(0.1, olasılıklar[indeks] - 0.1)
"   Köpek düşünüyor: 'Bu harika değildi...'")
# Olasılıkları normalleştirme
olasılıklar = normalize(olasılıklar)
"Mevcut öğrenme değerleri:", [round(p, 2) for p in
olasılıklar])

# Sonuç gösterimi
"\n Eğitim tamamlandı!")
for action, value in zip(actions, olasılıklar):

" {action}: {value:.2f}")

favorite_action =
actions[olasılıklar.index(max(olasılıklar))]
print(f"\n  " Köpeğin en sevdiği hareket: {favorite_action}
  ")
```

İpucu 1: Mevcut olasılıklar burada saklanır,
köpek bu olasılıkları kullanarak bir eylem seçer.

İpucu 2: Köpek ödüllendirildiğinde olasılık ne kadar artmalıdır?

Çözüm parçası 1: olasılıklar

Çözüm parçası 2: 0.2

5. İSTASYON: Pekiştirmeli Öğrenme — Ödül Yoluyla Öğrenme

GELİŞMİŞ (Bölüm I)

Kod:

```
print("Bitkinizi hayatta tutun!")
print("Eylemler: sulama, gübreleme, hiçbir şey yapma")
bitki_durumu = {
"su": 5, # 0-10
"besinler": 5,
"sağlık": 10
}
def evaluate_state():
su, besinler = bitki_durumu["su"], bitki_durumu["besinler"]
if 3 <= su <= 7 and 3 <= besinler <= 7:
return ____ # Mükemmel!
elif 1 <= water <= 9 and 1 <= nutrients <= 9:
return _____ # Tamam
else:
return -1 # Kötü
for day in range(7):
print(f"\n--- Gün {day+1} ---")
print(f"Durum: Su={plant_state['water']}/10, Besinler=
{plant_state['nutrients']}/10")

action = input("Ne yapıyorsunuz? (sulama/gübreleme/hiçbir
şey): ")
```

5. İSTASYON: Pekiştirmeli Öğrenme — Ödül Yoluyla Öğrenme

GELİŞMİŞ (Bölüm II)

Kod:

```
# Eylemi gerçekleştir
if action == "su":
plant_state["su"] = min(10, plant_state["su"] + 3)
elif action == "gübrele":
plant_state["besinler"] = min(10, plant_state["besinler"] +
3)
# Zaman geçiyor - durum azalıyor
plant_state["su"] = max(0, plant_state["su"] - 1)
plant_state["besinler"] = max(0, plant_state["besinler"] - 1)
# Ödülü hesapla
reward = evaluate_state()
print(f"Ödül: {reward} puan")

if plant_state["su"] == 0 or plant_state["besinler"] == 0:

print("   Bitki öldü!")
break

if plant_state["su"] > 0 ve plant_state["nutrients"] > 0:

print("    Bitki hayatta kaldı! Aferin!")
```

İpucu 1: İdeal su ve besin seviyeleri için puanlar (maksimum ödül).

İpucu 2: Kabul edilebilir seviyeler için puanlar, mükemmel değil ama yine de iyi.

Çözüm parçası 1: 2

Çözüm parçası 2: 1

İSTASYON 6: Evrişimsel Sinir Ağları — Yapay Zeka Görüntülerdeki Kenarları Nasıl Algılar?

Aylin fotoğrafına bakıyor.

ALARA: “Çizgiler, parlaklık ve kontrastlar görüyorum—bunlar benim kenarlarım! Bir şey değiştiğinde biliyorum: burada yeni bir nesne başlıyor.”

A**LARA**: “Bunu tespit etmek için küçük bir hile kullanıyorum. Her zaman iki komşu pikseli karşılaştırıyorum.
Eğer farklılarsa, bir kenar beliriyor. != operatörü ‘eşit değil’ veya ‘farklı’ anlamına geliyor.”

ALARA: “Bana gösterdiğin her resim daha iyi ‘görmeme’ yardımcı oluyor,” diye açıklıyor Alara.
“Tıpkı senin öğrendiğin gibi, dünyayı giderek daha doğru bir şekilde tanımlıyorum.”

TEMEL (Bölüm I)

Kod:

```
def count_edges(row):
edges = 0
for i in range(len(row)-1):
if row[i] ___ rowi+1]:
edges += 1
return edges
print("Alara'nın 4 satırlık görüntüsü")
# Kullanıcıdan 4 satır girin
image = []
for n in range(4):
row = input(f"{n+1} satırını girin (0 ve 1, örneğin 100101):
")
row = row[:6]
image.append(row)
```

İSTASYON 6: Evrişimsel Sinir Ağları — Yapay Zeka Görüntülerdeki Kenarları Nasıl Algılar?

TEMEL (Bölüm II)

Kod:

```
# Satır başına kenar sayısını hesapla
for i, row in enumerate(image):
edges = count_edges(row)
if edges > 0:
print(f"Row {i+1}: {edges} edges")
else:
print(f"   Row {i+1}: no edges")
```

Çözüm parçası: i=

İpucu: Bir değişim (0 → 1 veya 1 → 0) bir kenardır.

GELİŞMİŞ (Bölüm I)

Kod:

```
print("Kendi 4x4 resminizi oluşturun!")
boyut = 4
resim = []
# Giriş satırları
for i in range(boyut):
satır = input(f"{i+1}. satırı girin (sadece    ve   ,
örneğin       ): ")
satır = list(satır.ljust(boyut, "  "))[:boyut]

resim.append(satır)
print("\n Resminiz:")
for satır in resim:

print(" ".join(satır))
```

İSTASYON 7: Kural Tabanlı Doğal Dil İşleme —Bilgisayarlar Dili Nasıl Anlar?

Aylin, Lena'ya bir mesaj yazıyor: "Hey, bu harikaydı!"

ALARA: "Mutlu, kızgın veya nötr olup olmadığını anlayabiliyorum —buna duygu analizi deniyor!"

ALARA: "Mesajlarınız insan dilini daha iyi anlamama yardımcı oluyor," diyor Alara minnetle. "Birlikte öğreniyoruz—siz bana yapay zekayı öğretiyorsunuz, ben de insan iletişimini öğreniyorum."

TEMEL

Kod:

```
print("Alara ruh halinizi okuyor!")
text = input("Bir şey yazın (örneğin, … harika olan / … kötü
olan): ")
def understand(text):
if "great" in text:
return "   İyi bir ruh halindesiniz!"

elif "bad" in text:
return ____
else:
return "   Nötr."

print(understand(text))
```

İSTASYON 7: Kural Tabanlı Doğal Dil İşleme —Bilgisayarlar Dili Nasıl Anlar?

GELİŞMİŞ

Kod:

```
print("Dil modeli gibi kelime sayımı!")
text = input("Bir cümle yazın: ")
def count_words(text):
words = text.split()
return len(____)

print("Kelime sayısı:", count_words(text))
```

Çözüm özeti: kelimeler

İpucu: Tüm kelimeleri içeren nedir?

8. İSTASYON: Öneri Sistemleri —YouTube Nasıl Bilebilir?

Aylin tabletinde videolar izliyor.

ALARA: “Neleri sevdiğini hatırlıyorum ve benzer içerikler öneriyorum!”

ALARA: “Neleri sevdiğini ne kadar çok gösterirsen, o kadar iyi önerilerde bulunabilirim,” diye açıklıyor Alara. “Bu bir ortaklık.”

TEMEL

Kod:

```
print("Alara'nın video önerisi")
video = input("Ne izliyorsunuz? (ör. müzik videosu, hayvan
videosu): ")
def recommend(video):
if "music" in video:
return "   Benzer müzik videoları!"
else:

return ____
print(recommend(video))
```

Çözüm özeti: "Yeni bir şey deneyin!"

İpucu: Müzik videosu değilse, yeni bir şey öneriin…

8. İSTASYON: Öneri Sistemleri —YouTube Nasıl Bilebilir?

GELİŞMİŞ

Kod:

```
print("YOUTUBE ASİSTANINIZ")
print("Mükemmel önerinizi bulun!")

name = input("\nAdınız nedir?")
print(f"Merhaba {name}! Nelerden hoşlanıyorsunuz?")
favorite1 = input("Favori Konu 1: ")
favorite2 = input("Favori Konu 2: ")
user_data = {
name: [favorite1, favorite2],
"Lena": ["Spor", "Oyun"]
}
def recommendation(user_name):
if favorite1 in ["Müzik", "müzik"]:
return "   ÖNERİNİZ: En İyi 100 Şarkı Listesi Karışımı!"
elif favorite1 in ["Spor", "spor"]:
return "⚽ ÖNERİNİZ: En İyi Goller Derlemesi!"
aksi takdirde:

"  ÖNERİNİZ: Trend Videolar!" döndürün

print(f"\n{öneri(____)}")
```

İpucu: Hangi değişken adı içeriyor?

Çözüm parçası: ad

İSTASYON 9: Hava Tahmini — Rastgele Orman

Aylin saatini işaret ediyor.

AYLIN: “Dışarıda oynayabileceğimizi her zaman nasıl biliyorsun?”

ALARA: “Bu benim uzman ekibim! İçimdeki üç zeki zihin birbirine danışıyor – tıpkı rastgele bir orman gibi!”

AYLIN: “Vay! Yani aslında içinde üç farklı uzman mı var?”

ALARA: “Evet! Biri sıcaklığa, biri bulutlara,
ve biri de rüzgara bakıyor. Birlikte en iyi kararı veriyoruz!”

ALARA: “Her tahminde daha da doğru tahminler yapıyorum,” diyor Alara.
“Geri bildirimleriniz uzmanlarımı daha da geliştirmeme yardımcı oluyor.”

Rastgele Orman algoritmasının özelliği şudur: Gerçekte, Rastgele Orman algoritması rastgele seçilmiş veriler ve özellikler kullanarak birçok ağacı eğitir. Oylama sistemimiz bu "kolektif zeka" ilkesini simüle eder.

İSTASYON 9: Hava Tahmini — Rastgele Orman

TEMEL

Kod:

```
print("AYLIN'İN HAVA DURUMU UZMANLARI")
print("Üç uzman görüşüyor!")

sıcaklık = float(input("Sıcaklık: "))
bulutlar = input("Bulutlar (güneşli/bulutlu): ")
# Uzmanlara sorun
uzman1 = "Güneşli" eğer sıcaklık > 20 ise aksi halde
"Yağmurlu"
uzman2 = "Güneşli" eğer bulutlar == "güneşli" ise aksi halde
"Yağmurlu"
fikirler = [uzman1, uzman2]
karar = max(set(fikirler), key=fikirler.____)
print(f"Tahmin: {karar}")
```

İpucu: En sık ortaya çıkan görüşü sayın.

Çözüm parçası: sayın

İSTASYON 9: Hava Tahmini — Rastgele Orman

GELİŞMİŞ

Kod:

```
print("AYLIN'İN HAVA KOMİTESİ")
print("Üç yapay zeka uzmanı karar veriyor!")

sıcaklık = float(giriş("Sıcaklık: "))
bulutlar = giriş("Bulutlar (güneşli/bulutlu/yağmurlu): ")
rüzgar = giriş("Rüzgar (güçlü/zayıf): ")
# Üç uzman görüşlerini veriyor
uzman1 = "Güneş" eğer sıcaklık > 20 ise aksi halde "Yağmur"
uzman2 = "Güneş" eğer ____ == "güneşli" ise aksi halde
"Yağmur"
uzman3 = "Güneş" eğer rüzgar == "zayıf" ise aksi halde
"Yağmur"
görüşler = [uzman1, uzman2, uzman3]
karar = max(set(görüşler), key=görüşler.count)

print(f"Tahmin: {karar}")
```

İpucu: Hangi değişken bulut bilgilerini içeriyor?

Çözüm parçası: bulutlar

Epilog — İkinci Hediye

A few months later.
Birkaç ay sonra.

Aylin masasında oturuyor, bileğinde Alara ışıldıyor.

Mutfakta bulaşıklar şangırdayarak ses çıkarıyor—Sema Teyze ziyarete gelmiş ve börek hazırlıyor.

AYLIN: "Biliyor musun Alara," diyor Aylin gülümseyerek, "doğum günümde bana hediye edildiğinde yapay zekâ hakkında hiçbir fikrim yoktu. Şimdi... ilk küçük projem üzerinde çalışıyorum ve senin nasıl 'düşündüğünü' anlıyorum."
Alara gururla göz kırpıyor. "Sadece yapay zekâyı anlamakla kalmadın —aynı zamanda bana daha çok insan gibi 'düşünmeyi' de öğrettin."
O anda Sema Teyze küçük, paketlenmiş bir hediyeyle içeri giriyor.

SEMA TEYZE: "Canım" diyor nazikçe,
"Son aylarda ne kadar çok şey öğrendiğini gördüm.
Büyükannen, 'Saate bir ruh verdin' derdi."
Aylin paketi açıyor. İçinde basit bir defter var.
İlk sayfada, akıcı bir yazıyla şunlar yazılı:
"Kalbinle icat et. Cesaretle programla.
Alara gibi düşün, ama Aylin gibi hisset."

Aylin gülümsüyor.

AYLIN: "Bu en güzel hediye, Sema Teyze."

SEMA TEYZE: "Hayır," diyor Sema, göğsüne hafifçe dokunarak.

"En güzel hediye zaten burada." SON

DÜŞÜNME SORULARI

Take a moment to think about what you have learned in this module.
Answer the following questions thoughtfully.

- **Yapay Zekayı Anlamak:** Bu modülü tamamladıktan sonra yapay zeka hakkındaki anlayışınız nasıl değişti? Alara gibi yapay zeka sistemlerinin "düşünme" biçimiyle ilgili sizi en çok şaşırtan şey neydi?

- **Sinir Ağları ve Karar Verme:** 2. İstasyonda, sinir ağlarının karar vermek için ağırlıkları nasıl kullandığını incelediniz. Bir seçim yapmadan önce farklı faktörleri tarttığınız gerçek hayattan bir durum düşünebilir misiniz? Bu, yapay zekanın karar verme biçimine nasıl benziyor?

- **Etik Hususlar:** Modül boyunca yapay zeka ile ilgili etik sorular gündeme getirildi. Yapay zeka tasarlarken veya kullanırken etik konuları düşünmek neden önemlidir? Günlük hayatta yapay zeka ile ilgili bir etik endişe örneği verin.

- **Etkileşim Yoluyla Öğrenme:** Alara, Aylin'in etkileşimlerinden öğreniyor. Gerçek hayattaki yapay zeka sistemlerinin (öneri motorları veya sesli asistanlar gibi) kullanıcı etkileşimlerinden nasıl öğrendiğini düşünüyorsunuz? Bu tür öğrenmenin faydaları ve riskleri nelerdir?

- **Dünyanızda Yapay Zeka:** Günlük hayatınızda yapay zeka ile nerede karşılaşıyorsunuz? Bir örnek seçin (örneğin, sosyal medya, yayın hizmetleri, akıllı cihazlar) ve bu modülde öğrendiğiniz yapay zeka kavramlarından birini nasıl kullandığını açıklayın.

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

1. Alara nedir?
a) Aylin'in en iyi arkadaşı
b) Sıradan bir çalar saat
c) Akıllı saatteki bir yapay zeka
d) Aylin'in teyzesi

2. İstasyon 2'de, birden fazla ağırlıklı girdiye dayalı bir karar vermek için hangi programlama kavramı tanıtılıyor?
a) Karar Ağaçları
b) Takviyeli Öğrenme
c) Sinir Ağları
d) Kural Tabanlı Sınıflandırma

3. İstasyon 3, "Okul Güzergahı Dedektifi", esas olarak neyle ilgilidir?
a) Müzik önerileri
b) Karar ağaçları kullanarak güzergah planlaması
c) Görüntü tanıma
d) Dil işleme

4. İstasyon 4'te Alara bir şarkıyı nasıl "Pop" olarak sınıflandırır?
a) Gitar içeriyorsa
b) Yüksek sesliyse
c) Dans edilebilirse
d) Taylor Swift tarafından yapılmışsa

5. İstasyon 5'teki "Bitki Bakımı" alıştırması neyi simüle eder?
a) Görüntü Tanıma
b) Ödüller aracılığıyla Pekiştirmeli Öğrenme
c) Sinir Ağları
d) Karar Ağaçları

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

6. İstasyon 6'da Alara bir görüntüdeki kenarları nasıl algılar?
a) Komşu pikseller arasındaki farkları karşılaştırarak
b) Beyaz pikselleri sayarak
c) Sesli komutlar kullanarak
d) Kullanıcıya sorarak

7. İstasyon 7'de ("Doğal Dil İşleme"), bir metnin ruh hali nasıl tanınır?
a) Metnin uzunluğunu ölçerek
b) "Harika" veya "kötü" gibi belirli anahtar kelimeleri arayarak
c) Dilbilgisini kontrol ederek
d) Yazı tipini analiz ederek

8. İstasyon 8'deki öneri sistemi ne yapar?
a) Hava tahminleri oluşturur
b) Kullanıcının ilgi alanlarına göre videolar önerir
c) El yazısını tanır
d) Bir robotu labirentte yönlendirir

9. İstasyon 9'daki "Rastgele Orman" nasıl karar verir?
a) Sadece bir uzman karar verir
b) Birden fazla "ağaç"ın demokratik oylamasıyla
c) Rastgele seçim yoluyla
d) Derin sinir ağları yoluyla

10. Tüm kodlama yarışmalarında hangi programlama dili kullanılır?
a) JavaScript
b) Python
c) N1GL1B
d) C++

Modül 2: Yapay Zekalı Kaçış Odası Hesap Makinesiyle Hazine Avı

GİRİŞ

Akıllı telefonunuzun hangi videoları beğenebileceğinizi nasıl bildiğini hiç merak ettiniz mi?
Bu modülün amacı, Yapay Zekanın (YZ) nasıl "düşündüğünü", kararlar aldığını ve verilerden nasıl öğrendiğini açıklayan temel kavramları tanıtmaktır. İçerik, ilgi çekici, oyun tabanlı bir öğrenme deneyimi aracılığıyla matematik, mantık ve yaratıcılığı birbirine bağlayan basit bir teorik arka plan sunmaktadır.
Modülün sonunda, aşağıdaki gibi farklı beceriler kazanabileceksiniz:

- **Matematiksel Yeterlilik:** Temel aritmetik işlemleri gerçekleştirme, ortalamaları, yüzdeleri ve oranları anlama konusunda gelişmiş beceri. Problem çözmede hesap makinesi kullanma konusunda özgüven kazanma.
- **Eleştirel Düşünme ve Mantıksal Akıl Yürütme:** Öğrenciler ipuçlarını çözerek ve verilen verilere dayanarak kararlar alarak mantıksal düşünme becerilerini geliştireceklerdir. Matematiksel kavramları pratik senaryolara ve bulmacalara uygulayarak problem çözme yetenekleri güçlenecektir.
- **Karar Verme:** Öğrenciler, sınıflandırmayla ilgili görevleri çözerek (örneğin, hangi nesnenin belirli bir tanıma uyduğunu belirlemek), sayısal verilere ve mantığa dayalı olarak bilinçli kararlar verme pratiği yaparlar..

Modül Süresi	2 saat (1 saat öğrenme + 1 saat uygulamalı egzersizler)

EĞİTİM MATERYALLERİ

ÜNİTE 2.1 YAPAY ZEKÂ DESTEKLİ KAÇIŞ ODASININ ARKASINDA NE VAR?

Yapay Zeka, genellikle insan zekası gerektiren görevleri (örneğin akıl yürütme, problem çözme, öğrenme veya örüntü tanıma) makinelerin yerine getirme yeteneği olarak tanımlanabilir. Matematik, Yapay Zekanın özünde yer alır; olasılık, istatistik ve mantık, bilgisayarların sezgiden ziyade verilere dayalı kararlar almasına yardımcı olur. Veri ise Yapay Zekanın "yakıtı" görevi görür: Ne kadar kaliteli veri varsa, sistem örüntüleri tespit etmede ve tahminlerde bulunmada o kadar başarılı olur. Bu modülde, Yapay Zekanın gerçek sorunları yaratıcı yollarla çözmek için akıl yürütme, olasılık ve öğrenmeyi nasıl kullandığını keşfedeceksiniz.

Bu fikirleri anlamak için, Yapay Zekanın nasıl "düşündüğünü" temsil eden dört temel ilkeyi inceleyeceğiz:

- **Bayesian Ağları,** bilgilerin belirsiz olduğu durumlarda sonuçları tahmin etmeye yardımcı olur.
- **K-Means Kümeleme Algoritması,** benzer verileri gruplandırarak kalıpları ve ilişkileri bulmayı amaçlar.
- **Monte Carlo Yöntemleri,** olasılıkları tahmin etmek ve öngörülerde bulunmak için rastgele deneyler kullanan yöntemlerdir.
- **Hebbian Öğrenme,** tıpkı beynimizin yeni şeyler öğrenirken olduğu gibi, bağlantıların tekrar yoluyla nasıl güçlendiğini gösterir.

ÜNİTE 2.2 YAPAY ZEKÂ DESTEKLİ KAÇIŞ ODASININ ARKASINDA NE VAR?

Bu basit fikirler sayesinde, yapay zekanın sadece kodlamadan ibaret olmadığını, aynı zamanda mantıksal akıl yürütme, problem çözme ve yaratıcılıkla da ilgili olduğunu anlayacaksınız. Bu prensipler, yapay zekanın "yaparak öğrenme" yöntemini nasıl ortaya çıkardığını göreceğiniz Yapay Zeka Kaçış Odası'nın her aşamasında size rehberlik edecektir.

Bu dört fikir, yapay zekanın mantığı, matematiği ve yaratıcılığı birleştirerek nasıl öğrendiğini ve akıllı kararlar aldığını gösteriyor; bu da tam olarak bu yapay zeka kaçış odası macerasındaki bulmacaları çözerken yapacağınız şey!

"Yapay zekanın amacı insanları değiştirmek değil, insan yeteneklerini artırmaktır."
— Fei-Fei Li (Stanford Üniversitesi, Yapay Zeka araştırmacısı)

KAYNAKLAR

Yapay Zeka ve Karar Verme

- Russell, S., & Norvig, P. (2021). Yapay Zeka: Modern Bir Yaklaşım (4. baskı). Pearson

Olasılık, Simülasyon ve Öğrenme

Shalev-Shwartz, S., & Ben-David, S. (2014). Makine Öğrenmesini Anlamak: Teoriden Algoritmalara. Cambridge University Press. → Makine öğrenmesinin, kümelemenin ve verilerden öğrenmenin temelleri.

Khan Academy. (tarihsiz). İstatistik ve olasılık.
https://www.khanacademy.org/math/statistics-probability
→ Olasılık, ortalamalar ve simülasyonların basit ve anlaşılır açıklamaları.

ÖĞRENME PLATFORMLARINA BAĞLANTILAR

Yapay zekâ hakkında eğlenceli ve basit bir şekilde öğrenmeye devam etmek istiyorsanız, bu çevrimiçi platformlar harika bir başlangıç noktası! Yapay zekâ kaçış odasında olduğu gibi, bilgisayarların görüntüleri, sesleri veya desenleri nasıl tanıdığını oynayabilir, deneyebilir ve keşfedebilirsiniz. Her bağlantı, yaparak öğrenmeye devam ederken yapay zekânın gerçek hayatta nasıl çalıştığını anlamanıza yardımcı olan kısa etkinlikler içerir.

- Google Teachable Machine – Görüntüler, sesler veya pozlarla basit yapay zeka modellerini eğitmek için etkileşimli platform.
- https://teachablemachine.withgoogle.com/

- Khan Academy – Yapay Zeka ve Makine Öğrenimi – Günlük örnekler kullanarak temel yapay zeka kavramlarını açıklayan ücretsiz dersler.
- https://www.khanacademy.org/computing/computer-science

- Okyanuslar için Yapay Zeka (Code.org) – Öğrencilerin veri ve önyargı hakkında bilgi edinerek okyanusu temizlemek için bir yapay zekayı eğittiği eğlenceli etkileşimli etkinlik.
- https://studio.code.org/s/oceans

- Çocuklar için Makine Öğrenimi – Öğrencilerin yapay zeka modelleriyle deney yapmalarına ve bunları Scratch projelerinde kullanmalarına olanak tanıyan platform.
- https://machinelearningforkids.co.uk/

- IBM SkillsBuild for Students – Rozetler ve sertifikalarla temel yapay zeka ve veri okuryazarlığı kursları sunan ücretsiz çevrimiçi öğrenme platformu.
- https://skillsbuild.org/students

- Google Yapay Zeka Deneyleri – Yapay zekanın desenleri, sesleri ve hareketleri nasıl tanıdığını gösteren etkileşimli araçlar koleksiyonu.
  https://experiments.withgoogle.com/collection/ai

PRATİK ALIŞTIRMA

Yapay Zeka Kaçış Odasına Hoş Geldiniz!

Siz, 4 haneli dijital bir kilidin ardında saklı bir hazineyi arayan genç kaşiflerden oluşan bir ekibin parçasısınız. Kilidi açmak için, her biri Yapay Zekanın bir ilkesini temsil eden dört koruyucu tarafından korunan 4 odadan oluşan bir yoldan ilerlemeniz gerekiyor:

1. Olasılık (Bayes Ağları)
2. Kümeleme (K-Ortalama Kümeleme)
3. Simülasyon (Monte Carlo Yöntemleri)
4. Hebbian Öğrenimi

Göreviniz, her istasyondaki zorlukların üstesinden gelmek, sorunları çözmek ve ardından kilidi açacak kombinasyonu elde etmektir. Her istasyonda, aralarında kısa alıştırmalar bulunan bir açıklama ve kilidi açmak için gereken rakamlardan birini verecek üç zorluk seviyesine sahip son bir programlama görevi bulunur.

İstasyon 1 – Bayes Ağları

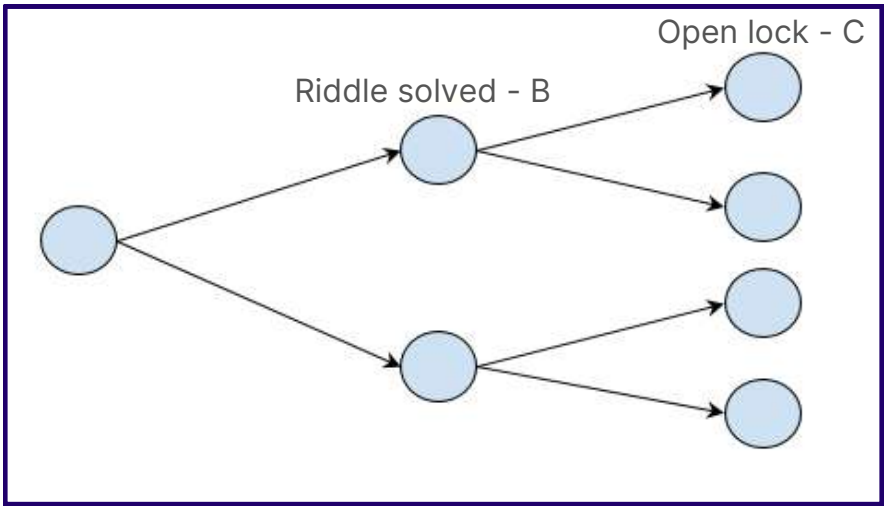
Bayes ağı, tüm bilgilerin kesin olarak bilinmediği, yani belirsizliğin olduğu durumlarda karar vermek veya tahminlerde bulunmak için olasılığı kullanan matematiksel bir modeldir.

Mini Görev 1 — Basit Kilit

Bir kaçış odasındaki sayısal kilidi açıp açamayacağınızı öğrenmek istediğinizi hayal edin.

- **Etkinlik A – İzler:** Birkaç gizli ipucu var. Ne kadar çok ipucu bulursanız, kilidi açmak o kadar kolaylaşır.
- **B Etkinliği – Bilmece:** Şifre kombinasyonunun sayılarını veren bir bilmece var. Ne kadar çok ipucu bulursanız, çözmesi o kadar kolaylaşır.
- **C Etkinliği – Kilit:** Bulmacayı çözüp kilidi açmak için gerekli sayıları girmeniz, bulduğunuz ipuçlarının sayısına bağlıdır.

Mini Görev 1 — Basit Kilit



FŞekil 2 Karar ve Olasılık Yolu: İpuçlarını Bulmaktan Kilidi Açmaya.
Kaynak: Monica Moreno

Not: Grafiğin yarısı eksik,
bu da ipuçlarını bulamama olasılığına karşılık geliyor.

Talimatlar:
İpuçlarını bulursanız (olasılık 0,8) ve bilmeceyi çözerseniz (ipuçlarını bulduysanız olasılık 0,9), kilidi açma olasılığınız nedir (bilmeceyi çözdüyseniz olasılık 0,95)?

Çözümü bulma yolu, çarpılması gereken bir dizi olasılıktan oluşmaktadır.

Cevap: $P(\text{Açık Kilit}) = 0.8 \times 0.9 \times 0.95$

Soru: Tüm olası yolları toplarsanız hangi sonucu elde edersiniz?

Cevap: 1, çünkü her yol olası bir senaryonun gerçekleşme olasılığını temsil eder. Tüm olası sonuçlardan birinin gerçekleşme olasılığı her zaman 1'dir.

Bayes Kuralı

Şimdiye kadar nasıl tahmin yapabileceğimizi gördük, peki ya sonucu inceleyip bir yolun izlenme olasılığını öğrenmek istersek ne olur?

Mini-Meydan Okuma 2 — Şartlı Koridor

Duvarında başka bir bilmece bulunan gizli bir koridor buluyorsunuz: “Bilmeceyi çözdüğünüz tüm zamanlar arasında, ipuçlarını ilk önce bulma olasılığınız nedir?”

Başka bir deyişle, C'nin B'ye doğru gitme olasılığını hesaplamak istiyoruz (bu, B'nin C'ye doğru gitme olasılığından farklıdır). Bayes Kuralını kullanmalısınız:

Talimatlar:
Öncelikle, B=✓ ve C=✓ olasılıklarını hesaplamalıyız, çünkü B=✓ ile biten tüm yollar ve bunların tümü C= ✓ ile biter. İşleri basitleştirmek için, zaten yapılmış olan işlemleri size vereceğiz: P(B=✓)= 0,76, P(C=✓) = 0,734.

Şimdi sadece Bayes kuralını şu formülle uygulamamız gerekiyor:

$$P(B=✓ \text{ if } C=✓) = [P(C=✓ \text{ if } B=✓) * P(B=✓)] / P(C=✓)$$

Cevap: $P(B=✓|C=✓) = [0.95 * 0.76] / 0.724 = 0.983$

1. Meydan Okuma - Olasılığın Koruyucusu

Dijital meşalelerle aydınlatılmış gizemli bir salon olan Olasılık Bekçisi Odası'na ulaştınız. Ortada, birçok değerli taşla kaplı eski bir sandık duruyor. Ardından, yaşlı bir adamın hologramı olan bekçi size şöyle diyor:

“Sadece belirsizliği anlayanlar sandığın altın içerip içermediğini keşfedebilecek. Olasılık anahtardır.”

Göreviniz, basitleştirilmiş bir Bayes Ağı modeli kullanarak, yalnızca görebildiklerinize dayanarak sandığın altın içerme olasılığını hesaplamaktır.

Senaryo: Bu ağır, parlak sandığın altın içerme olasılığı nedir? Cevabın ilk ondalık basamağı, kilidin ilk rakamı olacaktır.

- Parlak sandıkların %30'unun altın, %70'inin ise taş içerdiğini biliyorsunuz.
- Altın içeren sandıkların %80'i ağırdır.
- Taş içeren sandıkların ise %40'ı ağırdır.

1. Meydan Okuma - Olasılığın Koruyucusu

KOLAY SEVİYE: Bayes ağını bir kağıda çizin ve cevabı bulmak için hesap makinesi kullanın. İpucu: Sadece 2 olaya (A ve B) ve Bayes kuralına ihtiyacınız var.

ORTA SEVİYE: Bu eksik kodu, problemi çözmek için değiştirin. Sadece olasılıkları ve matematiksel sembolleri ekleyin.

TEMEL

```
KOD:
P_GOLD =
P_STONES =
P_HEAVY_IF_GOLD =
P_HEAVY_IF_STONES =

# AĞIR OLMA OLASILIĞINI HESAPLA
P_HEAVY = (P_GOLD “OPERATION” P_HEAVY_IF_GOLD) + (P_STONES
"OPERATION" P_HEAVY_IF_STONES)

# AĞIR OLDUĞU BILINDIĞINDE ALTIN OLMA OLASILIĞINI HESAPLA
P_GOLD_GIVEN_WEIGHED = (P_GOLD "OPERATION" P_WEIGHED_IF_GOLD)
"OPERATION" P_WEIGHED

print("Ağır olduğu bilindiğinde altın olma olasılığı:",
round(p_gold_given_weighed, 2))
```

ZORLUK SEVİYESİ: Bayes ağını çizin ve problemi çözmek için kendi kodunuzu yazın.

İstasyon 2 — K-Ortalama Kümeleme

K-Means, verileri benzerliklerine göre kategorilere ayıran bir algoritmadır.
Bunu anlamak için bir örneğe bakalım. Yılın neredeyse her günü sıcaklığı ölçtüğünüzü ve her noktanın hangi mevsime karşılık geldiğini bilmek istediğinizi hayal edin. Buna sınıflandırma problemi denir.

Mini Görev 1 — Koridordaki Kristaller

Bir sonraki odaya giderken yerde birkaç cam bardak bulacaksınız. Her kristalin üzerinde iki sayı yazılı: parlaklık ve saflık. Bu kristalleri özelliklerine göre iki ayrı gruba ayırın.

Talimatlar:

- İki eksen (x,y) çizin ve noktaları işaretleyin:
- [1,5], [4,1], [3.5,0], [2,4], [0.5, 5], [0.5, 5], [5, 2].
- İki grup veya küme (k=2) oluşturmak istediğinizi düşünün.
- Her grubun veya ağırlık merkezinin başlangıç merkezini nereye yerleştirirsiniz?
- Her kristali en yakın ağırlık merkezine atayın.

Cevap: Group 1: [1,5], [2,4], [0.5, 5], [0.5, 5]. Group 2: [4,1], [3.5,0], [5, 2]

K-Means, her bir grubun özelliklerinin ortalamasını temsil eden bir "merkezi" (ağırlık merkezi) olacak şekilde grup oluşturma sürecini otomatikleştirir. "K-Means Nasıl Çalışır?"

1. Küme sayısını seçin (k=n).
2. Düzlem üzerine her grup için bir tane olmak üzere k adet ağırlık merkezini rastgele yerleştirin.
3. Düzlem üzerindeki her nokta, en yakın ağırlık merkezine atanır.
4. Ağırlık merkezini yeniden konumlandırmak için her alanın "merkezini" hesaplayın.
5. Noktalar artık grup değiştirmeyene kadar işlemi tekrarlayın.

2. Meydan Okuma - Simyacı

İlk odayı geçtikten sonra, gizemli renklerde sıvılarla dolu birçok şişenin bulunduğu bir laboratuvara ulaşacaksınız. Sonra yaşlı bir simyacı size şöyle der:

"Genç maceracı, iksirlerimi ayırmadan buradan ayrılamazsın. Her şişe farklı bir sihir türüne ait: Şifa, Güç veya Görünmezlik. Eğer onları doğru şekilde gruplandırmama yardım edersen, kapıyı senin için açacağım.

Her kavanozun iki kimyasal özelliği vardır: pH ve sihirli enerji konsantrasyonu. Göreviniz, K-Means algoritmasını kullanarak kavanozları gruplandırmak ve böylece ikinci testi geçmektir. Oluşturduğunuz küme sayısı, kilidin ikinci basamağına karşılık gelecektir.

KOLAY SEVİYE: Eksik parametreleri doldurarak kavanozları gruplandırmak için aşağıdaki kodu kullanın. EKSTRA: Grup sayısını değiştirirken sonucun nasıl değiştiğini görün.

NOT: Öncelikle scikit-learn kütüphanesini kurmanız gerekiyor (pip install scikit-learn)

ORTA SEVİYE: Kolay seviyedeki kodu kullanın ve istediğiniz koordinatlara sahip yeni bir kavanoz (nokta) ekleyin. Sınıflandırmayı almak için kmeans.predict(nokta) fonksiyonunu kullanın. Bu kodu ekleyin. Sonucu görüntülemek için en sonda.

ZORLUK SEVİYESİ: Kodu sıfırdan oluşturun.

ZORLUK SEVİYESİ: Bayes ağını çizin ve problemi çözmek için kendi kodunuzu yazın.

2. Meydan Okuma - Simyacı

TEMEL

KOD:

```
FROM SKLEARN.CLUSTER IMPORT KMEANS # KMEANS KÜTÜPHANESI
IMPORT NUMPY AS NP
IMPORT MATPLOTLIB.PYPLOT AS PLT

# VERI: [PH, GÜÇ]
POTIONS = NP.ARRAY([
[2, 53], [6, 70], [3, 55], [3.1, 52], [10, 25],
[8, 84], [9, 80], [7.5, 82],
[10, 19], [10.5, 22], [4, 48], [9, 20],
])

# K-MEANS
KMEANS = KMEANS(N_CLUSTERS=2, RANDOM_STATE=0) # ALGORITMA
BILDIRIMI
kmeans.fit(potions) # K-means algoritmasını çalıştırarak
kümeleri sınıflandırır
```

ZORLUK SEVİYESİ: Bayes ağını çizin ve problemi çözmek için kendi kodunuzu yazın.

2. Meydan Okuma - Simyacı

ORTA

KOD:

```
plt.scatter(potions[:, 0], potions[:, 1], c=kmeans.labels_,
cmap="viridis", marker="o")
plt.scatter(
new_jar[:, 0], new_jar[:, 1],

c=[plt.cm.viridis(prediction[0] / (kmeans.n_clusters - 1))],

marker="x", s=150, label="new jar"

plt.xlabel("pH")
plt.ylabel("Sihir Gücü")
plt.title("Özel Şişe Sınıflandırması")
plt.legend()
plt.show()
```

ZORLUK SEVİYESİ: Bayes ağını çizin ve problemi çözmek için kendi kodunuzu yazın.

2. Meydan Okuma - Simyacı

ZOR

KOD:

```
`from sklearn.cluster import KMeans`
`import numpy as np`
`import matplotlib.pyplot as plt`

# Adım 1: Verileri tanımlayın (numpy ile [pH, enerji]
çiftlerinin listesi)
potions = np.array([

# verilerinizi buraya ekleyin
])

# Adım 2: Küme sayısını belirterek KMeans modelini oluşturun
kmeans = KMeans(n_clusters=2, random_state=0)

# Adım 3: Verilerinizle modeli eğitin (fit)
kmeans.fit(potions)

# Görselleştirme
plt.scatter(potions[:, 0], potions[:, 1], c=kmeans.labels_,
cmap="viridis", marker="o")
plt.xlabel("pH")
plt.ylabel("Sihirli Güç")
plt.title("İksir sınıflandırması ile K-Means")
plt.show()

# Adım 4: Sonucu yazdırın
print("Her kavanozun sınıflandırması:", kmeans.labels_)
```

İstasyon 3 — Monte Carlo Yöntemleri

Monte Carlo yöntemleri, aynı deneyi birçok kez simüle ederek ve o olayın sonuç olarak kaç kez gerçekleştiğini sayarak bir olayın gerçekleşme olasılığını tahmin etmenin bir yoludur.

MMini Görev 1 — Servet Çeşmesi

Bir süre yürüdükten sonra bir sonraki odayı bulamayınca kaybolmaya başlıyorsunuz. Sihirli bir çeşmenin bulunduğu küçük bir odaya varıyorsunuz ve yolunuzu bulmak için bir dilek dilemeye karar veriyorsunuz.

Taşın üzerindeki yazı şöyle:

“Eğer bir madeni parayı atarsanız, dileğinizin gerçekleşme olasılığı %50'dir.
Peki ya bunu birçok kez denerseniz?”

Talimatlar

- Hesap makinesini kullanarak on adet rastgele sayı üretin. Çift sayılar 'tura'yı, tek sayılar ise 'yazı'yı temsil eder.
- Madeni paranın tura gelme sayısını sayın ve toplam atış sayısına (N=10) bölün. Sonuç, olasılıktır.

Monte Carlo yöntemlerini kullanabilmek için deneylerde her zaman rastgelelik bulunmalıdır. Rastlantı olmasaydı, sonuçlarda hiçbir çeşitlilik olmazdı, bu da olasılıkları tahmin edemeyeceğimiz anlamına gelirdi.

3. Görev - Kader Zarlarının Bekçisi

Meşalelerle aydınlatılmış taş bir koridorda uzun süre yürüdükten sonra, üzerinde parşömenler, renkli taşlar ve bazı veriler bulunan yuvarlak bir masanın bulunduğu bir odaya ulaşırsınız.

Masanın arkasında, derin bir sese sahip, kapüşonlu bir iskelet olan Şans Muhafızı oturmaktadır. Üç zar alıp size şöyle der:

"Yolcu, bu oda güç veya zekâ ile açılmaz...
Ben sonsuz Monte Carlo oyunlarının muhafızıyım ve size bir meydan okuma sunuyorum.
Bu üç zarı atıp sayıları toplayacağım.
Seçim: 12'den büyük veya 12'ye eşit/küçük.
Doğru tahmin ederseniz ilerleyebileceksiniz,
ama yanlış tahmin ederseniz sonsuza dek burada kilitli kalacaksınız."

Hangi seçeneği seçeceğinize karar vermek için, olasılık bilginizi uygulamaya ve en iyi seçeneğinizi belirlemek için Monte Carlo yöntemlerini kullanmaya karar verirsiniz. En yüksek olasılığın ilk ondalık basamağı size kilidin üçüncü rakamını verecektir.

KOLAY SEVİYE: Aşağıdaki adımları izleyerek, N=10.000 deney için her bir durumda başarı olasılığını hesaplamanıza olanak tanıyan bir kod oluşturun.

1. Her sonucun kaç kez göründüğünü saymak için 2 değişken oluşturun.
   İpucu: Değişkenleri 0'a başlatmanız gerekiyor.
2. Üç rastgele sayı üretin ve bunları toplayın.
   İpucu: random.random() fonksiyonunu kullanmalısınız.
3. Toplamın 12'den büyük veya 12'ye eşit veya küçük olup olmadığını kontrol edin ve ilgili sayaca 1 ekleyin.
4. Bu işlemi 10.000 kez tekrarlayın.
   İpucu: Kodu bir "for" döngüsünün içine koymalısınız.
5. Olasılığı elde etmek için sayaçları N=10.000'e bölün ve en olası olanı seçin.

3. Görev - Kader Zarlarının Bekçisi

ORTA SEVİYE: N=10.000 iterasyon için deneyi simüle edecek kodu yazın (gerekirse kolay seviye adımlarını kullanabilirsiniz), ancak zar atıldığında her olası sonucun (3'ten 18'e kadar) olasılığını da hesaplamalısınız.

İpucu: Sayaçlar için bir sözlük kullanmalısınız. Tüm deneyleri çalıştırın ve sayacı toplayın. Son olarak, tüm deneylerden sonra, tüm değerleri N'ye bölün.

ZOR SEVİYE: Herhangi bir zar sayısı, tekrar sayısı ve eşik değeri için her olası sonucun olasılığını hesaplayan sıfırdan bir program oluşturun.

İpucu: N, num_dices ve threshold parametrelerini alan ve olası sonuçları anahtar, olasılıkları ise değer olarak içeren bir sözlük döndüren bir fonksiyon oluşturmalısınız.

EKSTRA (uzman seviyesi): Her değerin olasılıklarını bir histogramda (plt.bar) görüntülemeyi sağlayan olasılık dağılımının görselleştirilmesini ekleyin.

İstasyon 4 – Hebbian Öğrenimi

Hebbian öğrenme, "Birlikte ateşlenen nöronlar birlikte bağlanır" cümlesine dayanan sinir ağları için bir öğrenme kuralıdır. Başka bir deyişle:

- İki nöron aynı anda aktif hale gelirse, aralarındaki bağlantı güçlenir.
- Biri aktif olup diğeri aktif değilse, bağlantı zayıflar.

Mini Görev 1 — Fikirler Kütüphanesi

Son odaya giderken, üzerinde parlayan semboller bulunan iki heykelin olduğu karanlık bir koridordan geçiyorsunuz. Her "ışık" dediğinizde, iki heykel birlikte parlıyor. Ama "gölge" dediğinizde sadece biri parlıyor.

Hebb kuralına göre, "ışık" kelimesini birçok kez tekrarlarsanız bu iki heykel arasındaki bağlantıya ne olur?

a) Zayıflayacak
b) Aynı kalacak
c) Daha güçlü olacak

Mini Görev 1 — Fikirler Kütüphanesi

İki nöron ateşlendikten sonra aralarındaki bağlantı ağırlığını hesaplamak için Hebbian formülü kullanılır:

$$w_{final} = w_{initial} + (n_{together} \times \Delta w_{+}) + (n_{alone} \times \Delta w_{-})$$

Nerede:

- winitial = bağlantının başlangıç ağırlığı
- ntogether = her iki nöronun birlikte ateşlendiği sayı
- Δw+ = ortak aktivasyondan kaynaklanan ağırlık artışı
- nAlone = yalnızca bir nöronun ateşlendiği sayı
- Δw- = yalnızca bir nöron aktif olduğunda ağırlık kaybı

Mini-Meydan Okuma 2 — Bağlantı Ağırlıkları

Koridoru doğru şekilde aydınlatmak için, iki heykel (veya nöron) arasındaki aktivasyonu hesaplamanız gerekir. Bunu yapmak için Hebbian formülünü kullanmalısınız.

Talimatlar

- Bağlantının başlangıç ağırlığı 0,2'dir.
- Her ikisi birlikte açıldığında → +0,1
- Her seferinde sadece biri yandığında → −0,05
- Her iki ışık 3 kez birlikte yanıyor, ancak A ışığı 2 kez tek başına yanıyor

Cevap: $w_{final} = 0.2 + (3 \times 0.1) + (2 \times 0.05) = 0.6$

4. Meydan Okuma - Nöron Duvar Resmi

Sona çok yaklaştınız, ancak hâlâ aşmanız gereken son bir zorluk var. Son odada, birbirine bağlantı ağırlıklarını temsil eden ipliklerle birbirine bağlı 3 adet sihirli ışık (A, B, C) içeren bir duvar resmi bulacaksınız.
Gizemli bir ses size şöyle diyor:

"Son kapıyı açmak için hangi bağlantıların güçlendiğini keşfetmelisiniz.
Birkaç aktivasyondan sonra son ağırlıkları hesaplayın ve gizli deseni ortaya çıkarın."

Her bağlantı 0 ağırlıkla başlar. İki nöron birlikte ateşlenirse, bağlantı 0,2 artar; yalnızca biri ateşlenirse, ağırlık -0,07 azalır. Üç bağlantı vardır: AB, AC ve BC. Kilidin son rakamı, 4. turdan sonra AC bağlantısının ağırlığının son ondalık basamağına karşılık gelir. Tur başına aktivasyon sayısı:

Tur	Aktifleşmiş nöronlar
1	A,B
2	B,C
3	A
4	A, C
5	A, B, C

KOLAY SEVİYE: Sadece AB bağlantısı için hesaplamaları yapın,
her tur için bağlantı ağırlığı sonucunu gösteren bir tablo oluşturun.

ORTA SEVİYE: Tüm turlardaki tüm bağlantıların nihai ağırlıklarını hesaplayın.

ZOR SEVİYE: Üç nöronun da aynı anda ateşlenmesi durumunda tüm bağlantıların +0,2 bonus aldığını varsayarak, tüm turlardaki tüm bağlantıların nihai ağırlıklarını hesaplayın. Son turdaki en güçlü bağlantıyı belirleyin.

DÜŞÜNME SORULARI

Bu modülde öğrendikleriniz üzerine biraz düşünün. Aşağıdaki soruları dikkatlice yanıtlayın.

- **Yapay Zeka, Matematik ve Karar Verme:** Bu modülde, Yapay Zeka Kaçış Odası zorluklarını çözmek için matematik, olasılık ve mantık kullandınız. Bu etkinlik, yapay zekanın nasıl karar verdiğini anlamanıza nasıl yardımcı oldu? Hangi zorluk bunu en iyi anlamanıza yardımcı oldu?

- **Olasılık ve Simülasyon Yoluyla Öğrenme:** Bayes Ağları, K-Ortalama Kümeleme, Monte Carlo yöntemleri ve Hebbian Öğrenme gibi farklı yapay zeka yöntemlerini incelediniz. Bu fikirlerden birini kendi kelimelerinizle açıklayabilir ve kaçış odasında nasıl kullanıldığına dair bir örnek verebilir misiniz?

- **Tekrarlama ve Deneyim Yoluyla Öğrenme:** Bazı zorluklar, doğru cevaba ulaşmak için hesaplamaları, simülasyonları veya denemeleri tekrarlamayı gerektiriyordu. Bu, yapay zeka sistemlerinin verilerden ve deneyimlerden nasıl öğrendiğine nasıl benziyor? Yapay zekanın bu şekilde öğrendiği gerçek hayattan bir örnek düşünebilir misiniz?

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

1. Bayes Ağları bize esas olarak ne konuda yardımcı olur?
a) Benzer veri noktalarını gruplandırmak
b) Belirsizlik altında karar vermek
c) Rastgele örnekleme ile hesaplamaları hızlandırmak
d) Nöron bağlantılarını güçlendirmek

2. Parlak sandıkların %30'u altın içeriyorsa ve altın sandıkların %80'i ağırsa, ağır bir sandığın altın içerme olasılığını bulmak için hangi kuralı kullanırız?
a) Monte Carlo Simülasyonu
b) Hebbian Öğrenme
c) Bayes Kuralı
d) K-Ortalama Algoritması

3. K-Ortalama'daki "k" neyi temsil eder?
a) Veri noktalarının sayısı
b) Küme sayısı
c) Boyut sayısı
d) Olasılık sayısı

4. Noktaları en yakın merkeze atadıktan hemen sonra hangi adım gelir?
a) Merkez noktalarını tekrar rastgeleleştirme
b) Her kümenin yeni merkezlerini hesaplama
c) Bayes Ağı çizme
d) Zar kullanarak olasılıkları tahmin etme

5. Monte Carlo yöntemleri olasılık tahmininde neden faydalıdır?
a) Rastgeleliği tamamen ortadan kaldırırlar
b) Rastgelelik içeren birçok deneyi simüle ederler
c) Verileri kategorilere ayırırlar
d) Ağ bağlantılarını güçlendirirler
Send feedback

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

6. "Kader Zarı Bekçisi" mücadelesinde ne hesaplanıyor?
a) Hangi sandıkta altın var?
b) Bir iksir hangi kümeye ait?
c) Zar toplamlarının 12'den büyük veya küçük olma olasılığı
d) Son nöron bağlantı ağırlığı

7. "Birlikte ateşlenen nöronlar birlikte bağlanır" ifadesine göre, iki nöron birçok kez aynı anda aktifleştiğinde ne olur?
a) Bağlantı zayıflar
b) Bağlantı kaybolur
c) Bağlantı güçlenir
d) Hiçbir şey değişmez

8. İki nöron 3 kez birlikte ateşlenirse (her biri +0.1) ve sadece biri 2 kez ateşlenirse (her biri –0.05), başlangıç ağırlığı = 0.2 ise, son ağırlık nedir?
a) 0.35
b) 0.60
c) 0.10
d) 0.45

9. Aşağıdakilerden hangisi Bayes Ağlarını Monte Carlo yöntemlerinden en iyi şekilde ayırır?
a) Bayes ağları olasılık kurallarını kullanırken, Monte Carlo simülasyonları kullanır.
b) Her ikisi de farklı isimlerle aynı yöntemdir.
c) Bayes ağları rastgeledir, Monte Carlo deterministiktir.
d) Monte Carlo, Hebbian öğrenme gibi bağlantıları güçlendirir.

10. Yapay sinir ağlarında, Hebbian öğrenme esas olarak şunlara yardımcı olur:
a) Verileri gruplara ayırma
b) Birlikte aktifleşen nöronlar arasındaki bağlantıları güçlendirme
c) Belirsizlik içeren sonuçları tahmin etme
d) Binlerce rastgele simülasyon çalıştırma

Modül 3: Scratch Yapay Zeka ile Buluşuyor

GİRİŞ

Bu modülün amacı, Scratch programlama dilini tanıtmak ve Scratch'in programlama kavramlarını tanıtmak için nasıl kolay ve etkili bir araç olabileceğini anlamaktır.

Modülün sonunda, aşağıdaki gibi farklı beceriler kazanabileceksiniz:

- **Scratch'in programlama kavramlarını öğrenmek için nasıl erişilebilir bir platform sağladığını anlayın.**
- **Blok tabanlı programlamanın temel becerilerini ve hesaplamalı düşünme yeteneğini geliştirin.**
- **Yaratıcı Scratch projelerine yapay zeka eklentileri entegre edin.**

Modül Süresi	2 saat (1 saat öğrenme + 1 saat uygulamalı egzersizler)

EĞİTİM MATERYALLERİ

Scratch, görsel tabanlı güçlü bir programlama dilidir; yani, geleneksel programlama dillerinin aksine, Scratch sözdizimine veya satır tabanlı koda dayanmaz; mantık oluşturmak için blokları birbirine bağlayarak çalışır. Programlamaya giriş olarak Scratch kullanmanın birçok avantajı vardır. Bu modülde, Scratch'in tarihçesini ve çevrimiçi platformu kullanarak nasıl kod yazılacağını öğreneceksiniz. Modülün sonunda, Scratch uygulamalarına yapay zeka uzantıları eklemeyi öğreneceksiniz.

ÜNİTE 3.1 SCRATCH'E GİRİŞ

Scratch, MIT Media Lab tarafından geliştirilen güçlü bir görsel programlama dili ve çevrimiçi topluluktur. Karmaşık kod öğrenmeyi ve yazmayı gerektiren geleneksel programlama dillerinin aksine, Scratch, yapboz parçaları gibi birbirine geçen kod blokları kullanır. Bu konsept, deneyim seviyeleri ve kodlama bilgileri ne olursa olsun, programlamayı herkes için erişilebilir hale getirir.

SCRATCH NEDEN PROGRAMLAMA ÖĞRENMEK IÇIN HARIKA BIR ARAÇ?

Scratch kullanarak kodlama öğrenmenin en önemli avantajları, geleneksel olarak yeni kodlayıcılar için programlamayı zorlaştıran birçok engeli ortadan kaldırmasıdır.

Örneğin, Scratch'in geleneksel programlama dillerinden farklılıkları şunlardır:

- **Sözdizimi Hatası Yok:** Scratch'te kod blokları yalnızca mantıklı bir şekilde bağlanır ve bu da sinir bozucu sözdizimi hatalarını önler.
- **Anında Görsel Geri Bildirim:** Scratch, kodunuzun sonuçlarını sahnede anında görmenizi sağlar.
- **Alçak Zemin, Yüksek Tavan:** Scratch öğrenmesi kolay, ancak aynı zamanda karmaşık projeler de yaratabiliyor.
- **Yaratıcı Özgürlük:** Oyunlar, animasyonlar, etkileşimli hikayeler, simülasyonlar ve daha fazlasını oluşturun...
- **İşbirliğine Dayalı Topluluk:** Projeleri paylaşın, başkalarının çalışmalarını yeniden düzenleyin ve akranlarınızdan öğrenin.
- **Çapraz Platform Desteği:** Scratch, tabletler ve akıllı telefonlar da dahil olmak üzere her türlü cihazda, önde gelen tüm web tarayıcılarında çalışır.

RSCRATCH'IN GERÇEK DÜNYADAKI UYGULAMALARI

Scratch, sözdizimi olmayan blok tabanlı bir dil olabilir; ancak Python, JavaScript ve Java gibi diğer dillere doğrudan aktarılabilecek temel programlama kavramlarını öğretir.
Scratch, programlama kavramlarını şu şekillerde ele alır:

Programming Concept	In Scratch	Real-World Application
Sequences	Blocks stacked in order	Step-by-step
Loops	Repeat blocks	Automating repetitive
Conditionals	If-then blocks	Decision-making logic in applications
Variables	Data storage blocks	Storing and managing
Events	Blocks that trigger	User interaction
Parallelism	Multiple scripts running	Multi-threaded

ÜNİTE 3.2 TEMEL BLOK TABANLI PROGRAMLAMA BECERİLERİ

SCRATCH HESABI NASIL OLUŞTURULUR?

Hesap açmak basit ve ücretsizdir. Öncelikle https://scratch.mit.edu/ adresindeki Scratch web sitesini ziyaret edin. Ardından ana gezinme menüsünden ‘Scratch'e Katıl’ seçeneğini belirleyerek bir hesap oluşturun ve kayıt talimatlarını izleyin.

SCRATCH ARAYÜZÜNDE GEZINME

Scratch arayüzü, bir programın ekranda nasıl görüneceğini kontrol eder. Platformda kodlamaya başlayabilmek için bilmeniz gereken yedi farklı Scratch öğesi vardır. Bunlar şunlardır:

- Arayüz
- Sprite'lar
- Sahne
- Bloklar
- Kostümler
- Arka Planlar
- Sesler

Scratch arayüzü üç ana alana ayrılmıştır: Projelerin görüntülendiği Sahne (sağ üst); komut dosyaları oluşturmak için blokları sürükleyip birleştirebileceğiniz Kod Alanı (ekranın ortası); ve karakterleri veya sprite'ları ve arka planları ekleyip yönetebileceğiniz Sprite Alanı (sağ alt).

Arayüzün sol tarafında, programları görsel olarak oluşturmak için farklı kod blok kategorilerine sahip bir araç kutusu olan Blok Paleti bulunur. Özetlemek gerekirse, arayüzün ana unsurları şunlardır:

- Sprite Alanı (Sağ Alt): Projenizdeki tüm sprite'ları (karakterler ve nesneler) gösterir. Kodunu düzenlemek için bir sprite'a tıklayın.
- Kod Alanı (Orta): Seçilen sprite için komut dosyaları oluşturmak üzere blokları sürükleyip bağlayabileceğiniz çalışma alanı.
- Sahne (Sağ Üst): Sprite'larınızın performans sergilediği ve etkileşimde bulunduğu görsel çıktı alanı. İzleyicilerinizin gördüğü yer burasıdır.

BLOK KATEGORILERINI ANLAMAK

Scratch, kod bloklarını dokuz renk kodlu kategoriye ayırır. Her kategori programınızda belirli bir amaca hizmet eder; kategoriler şunlardır:

1. **Mavi - Hareket:** Mavi - Hareket: Hareket kategorisi, karakterinizin ekranda hareket etmesini sağlar. Karakterin ileri doğru hareket etmesini istiyorsanız, X'i 10 Kat Değiştir bloğunu kullanabilirsiniz.

2. **Mor – Görünüm:** Görünüm kategorisi, karakterlerin ve arka planların görünümünü değiştirmeye odaklanır. Bir Karakterin görünümünü değiştirmek için Kostüm Değiştirme Bloğu kullanabilirsiniz.

3. **Macenta – Ses:** Ses kategorisi, sprite'ların ses çıkarmasını/çalmasını sağlamaya odaklanır. Bir ses çalmak için 'Ses Çalana Kadar Çal' bloğunu kullanabilirsiniz.

4. **Sarı – Olaylar:** Tüm Scratch projelerinin bir olaya ihtiyacı vardır; bu, etkileşimler gerçekleştiğinde olayların meydana gelmesini sağlar.

5. **Açık Turuncu – Kontrol:** Kontrol kategorisi, bir karakterin ne kadar süreyle bir şey yapması gerektiği veya ne kadar süreyle hareketsiz kalması gerektiğiyle ilgilidir. Programınıza bir "1 Saniye Bekle" bloğu sürükleyip bırakırsanız, program başlamadan önce 1 saniye bekleyecektir.
6. **Turkuaz – Algılama:** Bu kategori, bir nesnenin başka bir nesneye temas edip etmediğini kontrol ederek bir eylemin gerçekleşmesini sağlamakla ilgilidir.
7. **Yeşil – Operatör:** Bu kontroller, toplama, çıkarma ve dize birleştirme gibi matematiksel ve mantıksal işlemleri gerçekleştirmek için kullanılabilir.
8. **Koyu Turuncu – Değişkenler:** Değişkenler, puanlar veya sağlık puanları gibi bilgileri depolamak ve yönetmek için kullanılır.
9. **Kırmızı - Bloklarım:** Bu kategoride kendi özel bloklarınızı oluşturabilirsiniz.

ÜNİTE 3.3 SCRATCH'TE PROJE OLUŞTURMA

Scratch'te bir proje oluşturmak için, öncelikle kütüphaneden bir karakter (sprite) ve bir arka plan ekleyerek başlayın.

Yeni Bir Sprite Ekleme

Scratch düzenleyicisinin sağ alt köşesinde sprite kontrollerini bulacaksınız:

- Bir Sprite Seçin: Dahili sprite kütüphanesine göz atın (hayvanlar, insanlar, nesneler, fantastik karakterler vb.)
- Boyayın: Çizim araçlarını kullanarak kendi sprite'ınızı oluşturun
- Sürpriz: İlham almak için rastgele bir sprite ekleyin
- Sprite Yükleyin: Bilgisayarınızdan bir resim dosyası içe aktarın.

Arka Planı Değiştirme
Arka plan, projeniz için sahneyi hazırlar. Arka plan simgesine (sahnenin sağ alt köşesinde) tıklayarak şunları yapabilirsiniz:
Arka Plan Simgesi:

- Önceden hazırlanmış arka planlardan birini seçin (dış mekan, iç mekan, uzay, soyut vb.).
- Arka plan düzenleyiciyi kullanarak kendi arka planınızı oluşturun.
- Bilgisayarınızdan bir resim yükleyin.

Birden fazla arka plana sahip olabilir ve "arka planı [isim]" bloklarını kullanarak bunlar arasında programatik olarak geçiş yapabilirsiniz. Bu, çok sahneli oyunlar veya hikayeler oluşturmak için kullanışlıdır.

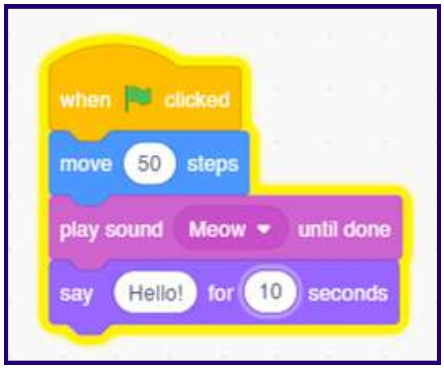
BIR KARAKTERI HAREKET ETTIRIN VE BIR EYLEMI TETIKLEYIN

1. **Adım – Olay Bloğu Ekle:** Olaylar kategorisine (sarı) tıklayın. 'Yeşil bayrağa tıklandığında' bloğunu kod alanına sürükleyin. Bu blok, sahnenin üzerindeki yeşil bayrağa tıkladığınızda programınızı başlatır.

2. A**dım – Hareket Ekle:** Hareket kategorisine (mavi) tıklayın. '10 adım hareket et' bloğunu sürükleyin ve olay bloğunun altına yerleştirin. Üzerine tıklayarak sayıyı 10'dan 50'ye değiştirin.

3. **Adım – Ses Efekti Ekle:** Ses kategorisine (macenta) tıklayın. 'Ses çal [miyav]' bloğunu sürükleyip bırakın ve hareket bloğunun altına bağlayın.

4. **Adım – Karakterin Konuşmasını Sağla:** Görünüşler kategorisine (mor) tıklayın. '2 saniye boyunca [Merhaba!] de' bloğunu sürükleyin ve komut dosyanıza ekleyin. "Merhaba!" ifadesini istediğiniz herhangi bir mesajla değiştirebilirsiniz.



Şekil 3 Hareket ve ses etkileşimi için Scratch blok mantığı:
MIT Media Lab tarafından geliştirilen Scratch

ÜNİTE 3.4 SCRATCH YAPAY İŞLEM UZANTILARINA GİRİŞ

SCRATCH'TE UZANTI NEDIR?

Uzantılar, Scratch'in yeteneklerini standart blok kategorilerinin ötesine genişleten ek modüllerdir. Uzantıları, sprite'larınıza metni sesli okumaktan yapay zeka yoluyla görüntüleri tanımaya kadar yeni güçler kazandıran özel araç setleri olarak düşünün.

Scratch 3.0, hesaplama özellikleri ekleyen yazılım uzantıları ve robotlar ve elektronik kitler gibi fiziksel cihazlara bağlanan donanım uzantıları içerir. Bu ünitede, projelerinize makine öğrenimi ve doğal dil işleme getiren yapay zeka destekli yazılım uzantılarına odaklanacağız.

SCRATCH'TE EKLENTI EKLEME

Uzantı eklemek çok kolay:

1. Blok alanının sol alt köşesinde "+" sembolü bulunan düğmeyi arayın.
2. Eklenti Kütüphanesini açmak için simgeye tıklayın.
3. Mevcut eklentilere göz atın ve projenize eklemek için birine tıklayın.
4. Yeni bloklar, kullanıma hazır olarak anında paletinizde görünecektir.

Çoğu yapay zeka eklentisi, yapay zeka işlemleri bilgisayarınızda değil, bulut sunucularında gerçekleştiği için aktif bir internet bağlantısı gerektirir. Bu eklentileri kullanmadan önce internete bağlı olduğunuzdan emin olun. Ayrıca, gizliliğe dikkat edin ve makine öğrenimi modellerini eğitirken kişisel veya hassas bilgileri yüklemekten kaçının

YARATICI PROJELERDE YAPAY ZEKA NEDEN VE NASIL KULLANILIYOR?

Yapay zeka, Scratch projelerinizin geleneksel programlama ile neredeyse imkansız olan şeyleri yapmasını sağlar:

Yapay Zeka Yeteneği	In Scratch	Real-World Parallel
Görüntü Tanıma	Web kamerasına çizdiğiniz veya gösterdiğiniz nesneleri tanımlayan bir oyun.	Google Fotoğraflar kişileri ve yerleri otomatik olarak etiketliyor.
Metin Sınıflandırması	A chatbot that understands whether messages are questions, statements, or commands	"Mesajların soru, ifade veya komut olup olmadığını anlayabilen bir sohbet robotu"
Konuşma Sentezi	Hikayeleri sesli okuyan veya oyun olaylarını anlatan karakterler.	GPS navigasyon sistemleri sesli yönlendirme sağlar.
Duygu Analizi	Nazik ve kaba sözlere farklı tepki veren sanal bir evcil hayvan.	GPS navigasyon sistemleri sesli yönlendirme sağlar.
Desen Tanıma	"Seslerin davul, piyano veya şarkı söyleme olup olmadığını belirleyen bir müzik uygulaması"	"Seslerin davul, piyano veya şarkı söyleme olup olmadığını belirleyen bir müzik uygulaması"

Yapay zeka, projeleri uyarlanabilir (girdiye göre farklı tepki veren), zeki (öğrenilmiş kalıplardan kararlar alan) ve etkileşimli (görüntü, metin ve ses gibi birden fazla girdi biçimini anlayan) hale getirerek yaratıcı olanakları genişletir.

BIR SPRITE'I HAREKET ETTIRIN VE BIR EYLEMI TETIKLEYIN

RESMI SCRATCH YAPAY ZEKA EKLENTILERI

- **Metinden Sese Dönüştürme:** Yazılı metni birden fazla ses ve dilde konuşulan kelimelere dönüştürür.
- **Çeviri:** Makine çevirisi kullanarak diller arasında metin çevirir.
- **Video Algılama:** Web kamerası aracılığıyla hareketi ve varlığı algılar (temel bilgisayar görüşü).

ÜÇÜNCÜ TARAF YAPAY ZEKA UZANTILARI

- **Çocuklar için Makine Öğrenimi:** Görüntü, metin, sayı ve ses tanıma için özel modeller eğitin
- **Teachable Machine:** Google'ın hızlı model eğitimi aracı, Scratch ile entegre edilebilir
- **Yüz Algılama:** Web kamerası aracılığıyla yüzleri ve yüz özelliklerini algılar.
- **Konuşma Tanıma:** Konuşulan kelimeleri metne dönüştürür (ses kontrolü)

ÜNİTE 3.5 YAPAY ZEKÂ UZANTISI: ÇOCUKLAR İÇİN MAKİNE ÖĞRENİMİ

IBM tarafından geliştirilen ve makine öğrenimini her yaştan öğrenci için erişilebilir kılan bir eğitim platformu olan Çocuklar İçin Makine Öğrenimi (ML4K), yapay zeka modellerini eğitmek için kullanıcı dostu bir arayüz sunar ve özel bloklar aracılığıyla Scratch ile sorunsuz bir şekilde entegre olur.

ML4K ile şu konularda modeller eğitebilirsiniz:

- Görüntüleri tanıma ve sınıflandırma (örneğin, kedileri köpeklerden ayırt etme veya elle çizilmiş şekilleri belirleme)
- Metni anlama ve kategorize etme (örneğin, mesajların övgü mü yoksa hakaret mi olduğunu tespit etme)
- Sayılardaki örüntüleri belirleme (örneğin, sayısal girdiye dayanarak sonuçları tahmin etme)
- Sesleri tanıma (örneğin, müzik notaları veya sesli komutlar arasında ayrım yapma)

MAKINE ÖĞRENIMI NEDIR?

Makine öğrenimi, bilgisayarların önceden programlanmış kuralları izlemek yerine örnekleri analiz ederek görevleri yerine getirmeyi öğrendiği yapay zekanın bir alt kümesidir. Örneğin, her köpek cinsi tanımını programlamak yerine, bilgisayara binlerce köpek resmi gösterirsiniz ve bilgisayar köpek cinsi özelliklerini tanımlayan özellikleri tanımayı öğrenir. Google Fotoğraflar gibi hizmetler resimlerinizi otomatik olarak bu şekilde etiketler.

ADIM ADIM: BIR MODELIN EĞITIMI VE KULLANIMI

ADIM 1: ÇOCUKLAR İÇIN MAKINE ÖĞRENIMINE ERIŞIM

Web tarayıcınızı açın ve şu adrese gidin:
https://machinelearningforkids.co.uk/

"Başla"ya tıklayın ve bu örnekte kayıt olmadan hesap oluşturmayı seçin.

Adım 2: Yeni Bir Proje Oluşturun Giriş yaptıktan sonra, "Projeler"e ve ardından "Yeni bir proje ekle"ye tıklayın.
Şunları yapmanız istenecektir:

- **Projenize isim verin: Açıklayıcı bir isim seçin (örneğin, "Duygu Dedektörü" veya "Geri Dönüşüm Ayırıcı")**
- **Tanıma türünü seçin: Metin, Sayılar, Görüntüler veya Sesler arasından seçim yapın**

Bu eğitimde, duygu sınıflandırıcı oluşturmak için "Metin"i seçin.

3. Adım: Etiketlerinizi (Kategorilerinizi) Tanımlayın

Makine öğrenimi modelleri, girdileri "etiketler" adı verilen kategorilere ayırır. Duygu sınıflandırıcı için iki etiket oluşturun:

- **Olumlu:** Mutlu, coşkulu veya nazik mesajlar
- **Olumsuz:** Üzgün, kızgın veya kaba mesajlar

Her kategori için "Eğit"e ve ardından "Yeni etiket ekle"ye tıklayın.

4. Adım: Eğitim Örnekleri Sağlayın
Model, örnekleri inceleyerek öğrenir.

Her etiket için en az 5-10 örnek ifade sağlayın:

Olumlu Örnekler	Olumsuz Örnekler
Bu inanılmaz!	Bunu beğenmiyorum.
Harika gidiyorsun!	Bu korkunç.
Yeni şeyler öğrenmeyi çok seviyorum.	Bu beni çok kızdırıyor.

İpucu: Ne kadar çok örnek verirseniz, modeliniz o kadar iyi performans gösterecektir.
Etiket başına en az on örnek hedefleyin ve ifade çeşitliliğine dikkat edin.

5. Adım: Modelinizi Eğitin
Yeterli örnek ekledikten sonra, projeye geri dönün ve "Öğren ve Test Et"i seçin, ardından "Yeni makine öğrenme modeli eğit" düğmesine tıklayın. Sistem, örneklerinizi analiz edecek ve yeni metni sınıflandırabilen bir model oluşturacaktır. Eğitim genellikle 1-3 dakika sürer. Bir ilerleme göstergesi göreceksiniz. Tamamlandığında, bir onay mesajı alacaksınız.

6. Adım: Modelinizi Test Edin
Modelinizi Scratch'te kullanmadan önce, ML4K içinde test edin; eğitim verilerinize dahil etmediğiniz bir ifade yazın (örneğin, "Bugün çok mutluyum!") ve "test et"e tıklayın. Model, girdinize en iyi uyan etiketi tahmin edecek ve bir güven yüzdesi gösterecektir.

Model hata yaparsa, eğitim bölümüne geri dönün
ve doğruluğu artırmak için daha fazla örnek ekleyin.

Adım 7: Scratch'e Bağlanın
ML4K, özel bir Scratch entegrasyonu sunar:

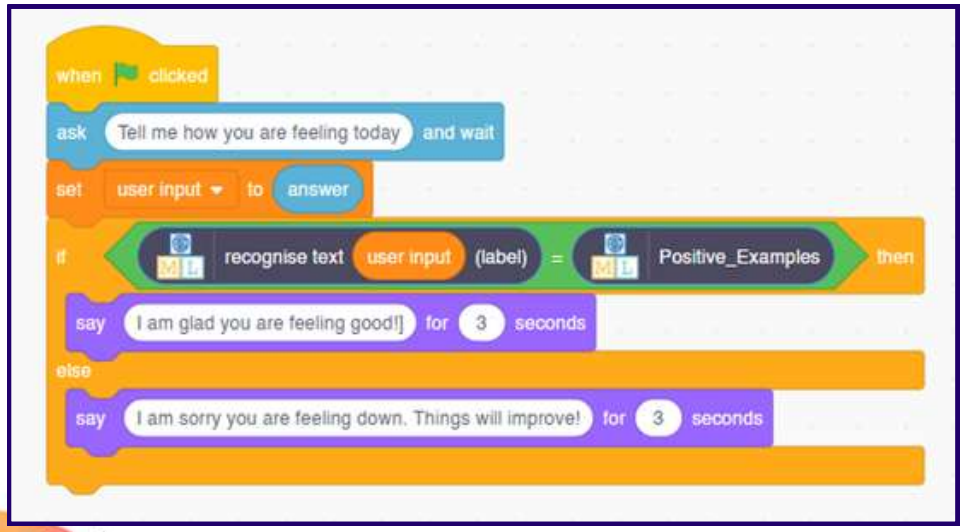
1. Proje navigasyonunda "Oluştur"a tıklayın.
2. "Scratch 3"ü seçin.
3. Bu, makine öğrenimi modelinizin özel bloklar olarak önceden yüklendiği değiştirilmiş bir Scratch düzenleyicisini açar.

Projenize özel yeni bloklar göreceksiniz, örneğin:

- Metni tanı [ ] (etiket)
- Metni tanı [ ] (güvenilirlik)

Adım 8: Scratch Projenizde ML Bloklarını Kullanın
Eğittiğiniz modeli kullanan basit bir Scratch projesi oluşturun: Öncelikle bir sprite ekleyin ("Robot" sprite'ını deneyin), ardından "kullanıcı girişi" adında bir değişken oluşturun ve şu komut dosyasını oluşturun:

Yeşil bayrağa tıklandığında [Bugün nasıl hissettiğinizi söyleyin] diye sorun ve bekleyin. [Kullanıcı girişi] değişkenini (cevap) olarak ayarlayın. Eğer <metni tanıyın (kullanıcı girişi) (etiket) = [Olumlu]> ise, 3 saniye boyunca [İyi hissettiğinize sevindim!] deyin, aksi takdirde 3 saniye boyunca [Üzgün hissettiğiniz için üzgünüm. İşler düzelecek!] deyin.



Şekil 4. Scratch'te temel bir olay-eylem zincirini gösteren kod bloklarının sırası: MIT Media Lab tarafından geliştirilen Scratch

Adım 9: Test Etme ve İyileştirme
Projenizi çalıştırın ve farklı mesajlar yazın. Yapay zekanın sınıflandırmasına göre sprite'ın nasıl tepki verdiğini gözlemleyin. Model metni yanlış sınıflandırırsa, ML4K'ye geri dönün, daha fazla eğitim örneği ekleyin, yeniden eğitin ve Scratch projenizi yenileyin.

ÜNİTE 3.6 YAPAY ZEKÂ UZANTISI: METİNDEN KONUŞMAYA DÖNÜŞTÜRME

Metinden Sese Dönüştürme (TTS), yazılı metni konuşulan sese dönüştüren yapay zeka destekli bir teknolojidir. Bu resmi Scratch eklentisi, bulut tabanlı konuşma sentezi kullanarak sprite'larınıza birden fazla dilde ve tonda gerçekçi sesler kazandırır.
TTS, projeleri daha erişilebilir (görme engelli kullanıcılar içeriği duyabilir), daha ilgi çekici (konuşulan diyalog, metin balonlarından daha dinamik hissettirir) ve daha çok yönlü hale getirir (projeler hikaye anlatabilir, talimatlar verebilir veya dil öğrenme araçları oluşturabilir).

METIN OKUMA EKLENTISININ EKLENMESI

Adım 1: Uzantı Kütüphanesini açın. Scratch düzenleyicisinde, blok paletinin sol alt köşesindeki mavi "Uzantı Ekle" düğmesine ("+" simgesiyle) tıklayın.

Adım 2: Uzantı kütüphanesinde ilerleyin ve "Metinden Sese Dönüştürme"ye (konuşma balonları olan bir hoparlör simgesi) tıklayın. Yeni bloklar, paletinizde yeni bir "Metinden Sese Dönüştürme" kategorisi altında görünecektir.

Adım 3: Blokları kodunuzda kullanın:

Bir cümle söyleyin: [hello] söyle bloğunu kod alanınıza sürükleyin. Blokun içine tıklayarak metni, karakterinizin söylemesini istediğiniz herhangi bir şeye değiştirebilirsiniz.

Sesi değiştirin: Farklı bir ses seçmek için [alto] ses ayarla bloğunu kullanın. Seçenekler arasında Alto, Tenor, Cıvıldama, Dev ve Yavru Kedi bulunur.

Dili değiştirme: Dili değiştirmek için, "dil ayarla" seçeneğini kullanın.

Şekil 5. Scratch'te konuşma için kod bloklarının sırası:
Scratch, MIT Media Lab tarafından geliştirilmiştir.

SCRATCH BLOKLARINI KULLANARAK ETKILEŞIMLI PROJELER OLUŞTURMA.

Metinden Sese Dönüştürme bloklarını diğer Scratch bloklarıyla birleştirerek etkileşimli projeler oluşturabilirsiniz. Örneğin, konuşmayı başlatmak için "Olaylar" kategorisindeki "yeşil bayrağa tıklandığında" olay bloğunu kullanabilirsiniz. Ayrıca, kullanıcının yazdıklarını karakterinizin tekrar etmesini sağlamak için "Algılama" kategorisindeki "sor ve bekle" bloğunu ve "konuş" bloğuyla birlikte "cevap" bloğunu kullanabilirsiniz.

TEMEL METIN-KONUŞMA BLOKLARI

Bu eklenti, konuşmayı kontrol etmek için aşağıdaki blokları sağlar:

Blok	İşlev	Örnek
konuş [metin]	Belirtilen metni sesli olarak okur.	"Merhaba, projemize hoş geldiniz!
sesi [ses] olarak ayarla	Ses karakterini değiştirir.	"Sesi [tenor] olarak ayarla (seçenekler: alto, tenor, cıvıldama, dev, kedi yavrusu)"
Dili [dil] olarak ayarla	Konuşma dilini değiştirir.	Dili [İspanyolca] olarak ayarlayın (23 dil desteklenmektedir)

METIN OKUMA SES SEÇENEKLERI

Karakterlerinizin kişiliğine uygun seslendirme seçeneklerinden bazılarını deneyebilirsiniz;

- Alto: Orta perdeli, nötr ses
- Tenor: Biraz daha yüksek perdeli ses
- Cikırtı: Yüksek perdeli, çocuksu ses
- Dev: Derin, gür ses
- Yavru kedi: Çok yüksek perdeli, sevimli ses

SCRATCH'TE METIN OKUMA ÖZELLIĞININ NASIL KULLANILABILECEĞINE DAIR ÖRNEKLER

- **Etkileşimli Hikayeler:** Karakterler olay örgüsünü anlatır ve diyalog kurar.
- **Eğitimsel Testler:** Erişilebilirlik için bilgisayar soruları sesli okur.
- **Dil Öğrenimi:** Farklı dillerle telaffuz pratiği.
- **Sanal Asistanlar:** Siri gibi sesli yanıtlar veren asistanlar.
- **Sesli Rehberler:** Anlatımlı bilgi içeren müze turları veya şehir rehberleri.
- **Dinamik Geri Bildirim:** Oyunlarda sözlü teşvik veya ipuçları.

KAYNAKLAR

Kitap, Wiki ve Dokümantasyon

Scratch Vakfı (MIT Medya Laboratuvarı): Platform dokümantasyonu, eğitim kaynakları ve arayüz açıklamaları https://scratch.mit.edu/

Çocuklar İçin Makine Öğrenimi (Dale Lane, IBM) 11 Şubat 2021: Platform eğitimleri, model eğitim metodolojisi ve entegrasyon kılavuzları. ISBN-13: 9781718500563

Scratch Wiki: Uzantılar ve blok açıklamaları için teknik dokümantasyon. https://scratch-wiki.info/

Yapay Zeka Scratch Kodlama Rehberi: Akıllı Sistemler Oluşturmaya Başlangıç Kılavuzu (Ciltli) – 11 Şubat 2025. ISBN-13: 979-8310433649

ÖĞRENME PLATFORMLARINA BAĞLANTILAR

Scratch Resmi Web Sitesi

MIT Media Lab tarafından geliştirilen resmi Scratch platformu.
Hesap oluşturun, projeler geliştirin, milyonlarca topluluk çalışmasını keşfedin, ve eğitimlere ve öğrenme kaynaklarına erişin.
https://scratch.mit.edu/

Çocuklar İçin Makine Öğrenimi

IBM tarafından özel makine öğrenimi modelleri (metin, resim, sayı, ses) eğitmek ve bunları Scratch ile entegre etmek için geliştirilen eğitim platformu.
Çalışma sayfaları, eğitimler ve önceden eğitilmiş modeller içerir.
https://machinelearningforkids.co.uk/
Send feedback

METIN OKUMA SES SEÇENEKLERI

Scratch ile Yapay Zeka Kavramlarını Yaratıcı Bir Şekilde Öğretmek (Codingal)
Yapay zeka kavramlarını Scratch aracılığıyla öğretmeye yönelik yaratıcı yaklaşımları, proje fikirlerini ve pedagojik stratejileri inceleyen blog yazısı.
https://www.codingal.com/coding-for-kids/blog/teach-kids-ai-codng-concepts-creatively-using-scratch/

Scratch Makine Öğrenimi Stüdyosu
Makine öğrenimi eklentilerini kullanan Scratch projelerinin derlenmiş koleksiyonu. İlham almak ve mevcut projelerden öğrenmek için keşfedin ve yeniden düzenleyin.
https://scratch.mit.edu/studios/3995548/

PRATİK ALIŞTIRMA

Amaç: Karakterlerinizin kişiliklerine uygun farklı sesler denemek için Metin-Konuşma eklentisini kullanarak bir proje oluşturmak.

Talimatlar: Metin girdisini sesli olarak okuyan ve metin girdisiyle konuşma çıktısını birleştirmenin gücünü gösteren bir karakter oluşturmak için aşağıdaki adımları izleyin.

Adım 1 - Projenizi Kurun:
Yeni bir Scratch projesi oluşturun veya mevcut bir projeyle devam edin.
Konuşacak bir karakter seçin – "Robot" veya "Büyücü" karakterleri iyi sonuç verir.

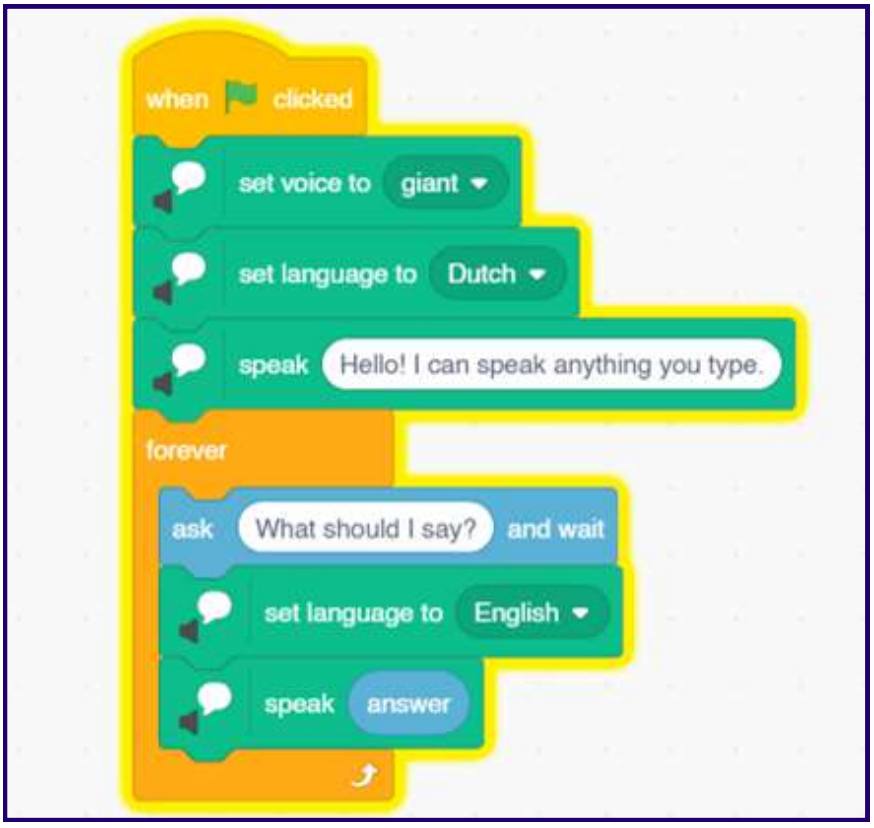
Adım 2 - Eklentiyi Ekleyin:
Yukarıda açıklandığı gibi Metin-Konuşma eklentisini ekleyin.

Adım 3 - Komut Dosyasını Oluşturun ve Karakteriniz için bu komut dosyasını oluşturun:

yeşil bayrağa tıklandığında
└── sesi [dev] olarak ayarla
└── konuş [Merhaba! Yazdığınız her şeyi okuyabilirim.]
└── sonsuza dek
└── sor [Ne demeliyim?] ve bekle
└── konuş (cevapla)

4. Adım - Projenizi Test Edin:

Yeşil bayrağa tıklayın ve istendiğinde mesajlar yazın. Karakteriniz mesajları sesli olarak okuyacaktır. Ses ayarı ve dil ayarı bloklarını değiştirerek farklı sesler ve diller deneyin.



Şekil 6. Scratch'te kod bloklarının sıralaması: MIT Media Lab tarafından geliştirilen Scratch

DÜŞÜNME SORULARI

Bu modülde öğrendikleriniz hakkında derinlemesine düşünmek için zaman ayırın. Aşağıdaki soruları dikkatlice yanıtlayın:

1. Yansıma Sorusu: Scratch'te yapay zekayı kullanmanın hangi yönü sizi en çok şaşırttı? Beklediğinizden farklı bir şey oldu mu? Projenizden belirli örneklerle açıklayın.

2. Yansıma Sorusu: Makine öğrenimini bir oyunda veya etkileşimli bir hikayede nasıl kullanırdınız? Belirli bir proje fikrini tanımlayın ve yapay zekanın bunu geleneksel programlamaya kıyasla nasıl daha ilgi çekici veya zeki hale getireceğini açıklayın.

3. Yansıma Sorusu: Makine öğrenimi modelinizi eğitirken veya yapay zeka uzantılarını entegre ederken hangi sorunlarla veya zorluklarla karşılaştınız? Bunları nasıl çözdünüz? Çözemediğiniz sorunlarla karşılaştıysanız, bir sonraki adımda hangi stratejileri denerdiniz?

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

1. Scratch'i en iyi tanımlayan ifade hangisidir?

a) Profesyonel yazılım mühendisleri tarafından kullanılan metin tabanlı bir programlama dili

b) Kodlamayı yeni başlayanlar için erişilebilir kılan görsel blok tabanlı bir programlama dili

c) Makine öğrenimi modelleri oluşturmak için bir yapay zeka hizmeti

d) Çocuklara programlama öğretmek için bir donanım robot kiti

2. Scratch'te blok tabanlı programlamanın en önemli faydalarından biri nedir?

a) Öğrencilerin sözdizimini ezberlemesini gerektirir, bu da hafıza becerilerini geliştirir

b) Bloklar yalnızca mantıksal olarak doğru şekillerde bağlanır, bu da sözdizimi hatalarını önler

c) Yalnızca metin tabanlı kodlamaya odaklanır, bu da öğrenenler için daha zordur

d) Yalnızca daha önce programlama deneyimi olan kişiler tarafından kullanılabilir

3. Kodu 10 kez tekrarlayan bir döngü oluşturmak için hangi Scratch blok kategorisini kullanırsınız?

a) Hareket

b) Olaylar

c) Kontrol

d) Operatörler

4. Scratch projenizde Çocuklar İçin Makine Öğrenimi modelini kullanmadan önce ne yapmalısınız?

a) Hiçbir şey - eklenti, tüm amaçlar için çalışan önceden eğitilmiş bir modelle birlikte gelir.

b) Modelin tanımasını istediğiniz her kategori için eğitim örnekleri sağlayın, ardından modeli eğitin.

c) Makine öğrenimini etkinleştirmek için özel bir donanım cihazı satın alın.

d) Bir makine öğrenimi algoritması oluşturmak için Python kodu yazın, ve bunu Scratch'e aktarın.

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

5. Aşağıdakilerden hangisi bu modülde bahsedilen yapay zeka ile ilgili bir yetenek DEĞİLDİR?
a) Görüntü sınıflandırması yoluyla nesneleri veya görüntüleri tanıma
b) Tekrar döngüsü kullanarak bir sprite'ın dans rutini gerçekleştirmesini sağlama
c) Konuşma tanıma yoluyla konuşulan kelimeleri metne dönüştürme
d) Bir proje içinde İngilizce'den İspanyolca'ya metin çevirme

6. Bir sprite'ın bir hikayeyi sesli olarak anlatmasını sağlamak için hangi Scratch eklentisini kullanırsınız?
a) Video Algılama
b) Çeviri
c) Metinden Sese Dönüştürme
d) Çocuklar İçin Makine Öğrenimi

7. Bir makine öğrenimi modeli oluştururken birden fazla eğitim örneği sağlamak neden önemlidir?
a) Model 100'den az örnekle çalışamaz
b) Daha fazla örnek, modelin kalıpları daha doğru bir şekilde öğrenmesine ve yeni girdileri daha güvenilir bir şekilde sınıflandırmasına yardımcı olur
c) Eğitim örnekleri yalnızca dekorasyon amaçlı kullanılır ve performansı etkilemez
d) Birden fazla örnek modeli yavaşlatarak daha doğru hale getirir

8. Çocuklar İçin Makine Öğrenimi ile Metinden Sese Dönüştürmeyi birleştirmenin avantajı nedir?
a) Projenin daha hızlı çalışmasını sağlar
b) Girdileri anlar ve yanıtları iletir
c) Yazmanız gereken kod miktarını azaltır
d) Projenin internet olmadan çevrimdışı çalışmasına olanak tanır

Module 4: Python Game Programming (pygame / Arcade libraries) with AI

INTRODUCTION

Bu kısa ve başlangıç seviyesine uygun modülde, sadece bir oyuncu değil, bir oyun yaratıcısı olacaksınız. Python ve Pygame kullanarak küçük bir video oyunu tasarlamayı ve programlamayı öğrenecek ve oyun karakterlerinizin basit Yapay Zeka (YZ) biçimleriyle nasıl akıllıca davrandığını keşfedeceksiniz. Daha önce hiç kodlama yapmadıysanız endişelenmeyin; her şey adım adım açıklanacaktır. Küçük bir oyun penceresi oluşturarak ve ekranda bir kareyi hareket ettirerek başlayacak ve daha sonra oyunun yaptıklarınıza nasıl tepki vereceğini, örneğin bir düşmanın hareketlerinizi fark etmesini veya yaklaştığınızda rengini değiştirmesini nasıl sağlayacağınızı keşfedeceksiniz. Bu modül sadece kod çalıştırmakla ilgili değil, aynı zamanda anlamak ve denemekle de ilgilidir. Küçük değişikliklerin farklı davranışlar yaratabileceğini görmek için örnekleri test edecek ve değiştireceksiniz. Bunu yaparak, oyunların gerçek dünyadaki basit YZ sistemleri gibi nasıl "düşündüğünü" ve tepki verdiğini öğreneceksiniz. Modülün sonunda, aşağıdaki gibi farklı beceriler kazanabileceksiniz:

- Pygame veya Arcade kütüphanelerini kullanarak etkileşimli 2D oyunlar oluşturun.
- Oyun karakterlerinin oyuncu eylemlerine tepki verebilmesi için yapay zeka davranışları ekleyin.
- Etkileşimli ve duyarlı oyunlar tasarlamak için hesaplamalı düşünme yöntemini kullanın.
- Oyun kodunu deneyin, test edin ve değiştirin;
  küçük değişikliklerin davranışı nasıl etkilediğini gözlemleyin.
- Yapay zekanın gerçek hayatta nasıl kullanıldığı üzerine düşünün.

Modül Süresi	**2 saat (1 saat öğrenme + 1 saat uygulamalı egzersizler)**

EĞİTİM MATERYALLERİ

ÜNİTE 4.1 PYGAME'E GİRİŞ

Kodlamaya başlamadan önce, hangi araçlarla çalışacağınızı ve bunların ne işe yaradığını bilmek önemlidir:

Python 3 – Bilgisayara komut vermek için kullanacağınız programlama dili. Okullarda, üniversitelerde ve şirketlerde programlama, veri analizi ve yapay zeka için yaygın olarak kullanılır.

Pygame – Basit 2 boyutlu oyunlar oluşturmanıza yardımcı olan ücretsiz bir Python kütüphanesi. Oyun penceresi açmak, şekiller çizmek, renkleri göstermek ve klavye veya fareye tepki vermek için hazır fonksiyonlar sağlar.

Kod editörü (IDLE, Thonny veya VS Code) – Python kodunuzu yazacağınız, kaydedeceğiniz ve çalıştıracağınız program. Bu modülde, IDLE'ı (Python ile birlikte gelir) veya öğretmeninizin önerdiği başka bir editörü kullanabilirsiniz.

Bu araçlar adım adım tanıtılacak, böylece programlamayla ilk kez tanışıyor olsanız bile bunları kurup etkinlikleri kendi başınıza tamamlayabileceksiniz.

Amaç: Bir oyun penceresi açmayı, renkleri görüntülemeyi ve klavyeyle bir kareyi hareket ettirmeyi öğrenmek.

Adım 1 – Python 3'ü Kurun ve Programlama Ortamınızı Açın
Pygame'i kullanmaya başlamadan önce, Python 3'ün
bilgisayarınızda kurulu olduğundan emin olmalısınız.
Python 3'ü Kurun

Resmi Python web sitesine gidin: https://www.python.org/downloads/.
Sisteminiz için (Windows, macOS veya Linux) sürümü indirin.
Kurulum sırasında, "Python'ı PATH'e Ekle" seçeneğini işaretleyin
ve ardından "Şimdi Kur" düğmesine tıklayın.

- Kurulum tamamlandıktan sonra bilgisayarınızı yeniden başlatın (isteğe bağlı ancak önerilir).

BÖLÜM 4.2 KURULUMUNUZU DOĞRULAYIN

Komut İstemi'ni (Windows) veya Terminal'i (macOS/Linux) açın ve şunu yazın:

python --versiyonu yada **py --versiyonu**

Eğer Python 3.13.3 gibi bir şey gösteriyorsa, her şey hazır demektir.

ÜNİTE 4.3 BİR DÜZENLEYİCİ KURUN VEYA IDLE KULLANIN

Python kodu yazmak için şunlardan birini kullanabilirsiniz:

IDLE (Python ile birlikte kurulu gelir)
Thonny (basit başlangıç seviyesi IDE) – https://thonny.org3
VS Code (gelişmiş seçenek) – https://code.visualstudio.com

IDLE'ı açmak için:

Windows'ta: Başlat → IDLE (Python 3.x)
macOS'ta: Launchpad'de "IDLE" arayın
Linux'ta: terminalde idle3 komutunu çalıştırın

3. Kodu Yazın veya Yapıştırın
Bu modüldeki kodu yeni dosyaya kopyalayın.
Örneğin:

KOD :

```
import pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
pygame.display.set_caption("My First Game")
running = True
while running:
   for event in pygame.event.get():
       if event.type == pygame.QUIT:
           running = False
   screen.fill((200, 220, 255))
   pygame.display.flip()
pygame.quit()
```

4. Dosyayı Kaydedin
Dosya → Farklı Kaydet... seçeneğini seçin.
Dosyanıza bir ad verin (örneğin my_first_game.py).

5. Programı Çalıştırın
F5 tuşuna basın veya Çalıştır → Modülü Çalıştır'a gidin.
Yeni bir pencere açılacaktır - eğer mavi bir arka plan görüyorsanız, her şey çalışıyor demektir.

ÜNİTE 4.4 TEMEL BLOK TABANLI PROGRAMLAMA BECERİLERİ

Bu kodu kopyalayıp çalıştırın:

KOD :

```
import pygame # 1. pygame kütüphanesini içe aktarın
pygame.init() # 2. Tüm pygame modüllerini başlatın
(başlatın)
screen = pygame.display.set_mode((800, 600)) # 3. Bir oyun
penceresi oluşturun (genişlik 800, yükseklik 600)
pygame.display.set_caption("İlk Oyunum") # 4. Oyun
penceresine bir başlık verin
running = True # 5. Oyunun çalışmasını sağlamak için bir
değişken oluşturun
while running: # 6. Ana oyun döngüsünü başlatın (tekrar
tekrar çalışır)
for event in pygame.event.get(): # 7. Tüm olayları kontrol
edin (örneğin, tuş basmaları veya fare tıklamaları)
if event.type == pygame.QUIT: # 8. Kullanıcı "kapat"
düğmesine tıklarsa…
running = False # 9. …oyun döngüsünü durdurun

screen.fill((200, 220, 255)) # 10. Ekranı açık mavi bir
renkle (RGB değerleri) doldurun
pygame.display.flip() # 11. Yeni çerçeveyi göstermek için
pencereyi güncelleyin
pygame.quit() # 12. Oyunu kapatın ve güvenli bir şekilde
çıkın
Send feedback
```

ÜNİTE 4.5 BİR KAREYİ HAREKET ETTİRME

Şimdi bir şeyler hareket ettirelim. Aşağıdaki kodu kopyalayın ve test edin:

BÖLÜM I

KOD:

```
import pygame # 1. pygame kütüphanesini içe aktarın
pygame.init() # 2. pygame'i başlatın
# 3. Pencereyi oluşturun
screen = pygame.display.set_mode((800, 600))
# 4. Bir başlık ekleyin
pygame.display.set_caption("Hareket Eden Kare")
x = 400 # 5. Karenin X konumu (yatay)
y = 300 # 6. Karenin Y konumu (dikey)
speed = 1 # 7. Her seferinde kaç piksel hareket eder

running = True # 8. Oyunu çalışır durumda tutun

while running: # 9. Oyun döngüsünü başlatın

for event in pygame.event.get(): # 10. Olayları kontrol edin
if event.type == pygame.QUIT: # 11. Kullanıcı "X"e tıklarsa,
durdurun
Çalışıyor = Yanlış
Geri bildirim gönder
```

ÜNİTE 4.6 BİR KAREYİ HAREKET ETTİRME

Şimdi bir şeyler hareket ettirelim. Aşağıdaki kodu kopyalayın ve test edin:

```
BÖLÜM II

KOD:
   tuşlar = pygame.key.get_pressed() # 12. Hangi tuşların
basılı olduğunu kontrol et
eğer tuşlar[pygame.K_LEFT]:

x -= hız # 13. Sola hareket et
eğer tuşlar[pygame.K_RIGHT]:
x += hız # 14. Sağa hareket et
eğer tuşlar[pygame.K_UP]:
y -= hız # 15. Yukarı hareket et
eğer tuşlar[pygame.K_DOWN]:
y += hız # 16. Aşağı hareket et
ekran.fill((200, 220, 255)) # 17. Arka planı boya
pygame.draw.rect(ekran, (0, 0, 255), (x, y, 50, 50)) # 18.
Mavi kareyi çiz
pygame.display.flip() # 19. Ekranı güncelle
pygame.quit() # 20. Oyunu kapat
Send feedback
```

Neler oluyor:
x ve y, karenin ekrandaki konumunu tanımlar.
Ok tuşları bu değerleri değiştirerek kareyi hareket ettirir.
Döngü, saniyede birçok kez çizimi tekrarlayarak,pürüzsüz bir hareket oluşturur.

Şunu deneyin:
Hızı = 1'den hız = 5'e değiştirin. Ne olur?
Kareyi farklı bir renge boyayın, örneğin
(255, 0, 0) kırmızı için.
Pygame.display.flip() fonksiyonunu kaldırırsanız ne olacağını düşünüyorsunuz?
Düşünme Sorusu: Bilgisayar, her tuşa bastığınızda karenin yeni konumunu nasıl biliyor?

Öğrenme Kontrol Noktası
Bu noktada şunları yapabilmelisiniz:
Pygame penceresi açmak.
Oyunun çalışmasını sağlamak için oyun döngüsünü kullanmak.
Hareket eden bir nesneyi klavye ile kontrol etmek.

Sonraki bölümde, oyununuzun hareketlerinize tepki vermesi için Yapay Zeka ekleyeceksiniz.

ÜNİTE 4.7 OYUNLARDA BASİT YAPAY ZEKÂ (REAKTİF DÜŞMAN)

Bu modülde, basit kural tabanlı oyun yapay zekasını kullanacağız: bilgisayar, oyuncuyu algılamak, ne yapacağına karar vermek ve buna göre hareket etmek için önceden tanımlanmış kuralları izler; verilerden öğrenme yapmaz. Bu, düşmanların oyuncuyu kovaladığı, ondan kaçındığı veya mesafeye veya konuma bağlı olarak oyuncuya tepki verdiği birçok klasik video oyununa benzer.

Amaç: Basit bir yapay zeka (YZ) biçimi ekleyerek oyununuzun oyuncunun yaptıklarına nasıl tepki vereceğini öğrenmek. Oyuncu yaklaştığında rengi değişen bir düşman yaratacaksınız.

Adım 1 – Fikri Anlamak

Kodlamaya başlamadan önce, bir oyunda yapay zekanın ne anlama geldiğini düşünün. Bu durumda, yapay zeka
öğrenme veya veriyle ilgili değil; bilgisayarın oyuncunun yaptıklarına tepki vermesiyle ilgilidir.

Basit bir kural kullanacaksınız: Oyuncu yakınsa, düşman renk değiştirir. Uzaksa, aynı kalır. Bu, yapay zekanın en basit biçimlerinden biri olan koşullu davranışın bir örneğidir.

Adım 2 – Kod Örneği: Tepkisel Düşman
Aşağıdaki kodu kopyalayıp çalıştırın:

BÖLÜM 1

KOD:

```
import pygame, math # 1. Oyun ve matematik kütüphanelerini
içe aktarın
pygame.init() # 2. Tüm Pygame fonksiyonlarını başlatın
(başlatın)
screen = pygame.display.set_mode((800, 600))
# 3. Bir oyun penceresi oluşturun (800x600)
pygame.display.set_caption("Basit Yapay Zeka - Tepkisel
Düşman")
# 4. Başlık
player = pygame.Rect(100, 100, 50, 50)
# 5. Oyuncu (mavi kare)
enemy= pygame.Rect(500, 300, 50, 50) # 6. Düşman (kırmızı
kare)
speed = 5 # 7. Oyuncu hızı
react_distance = 120
# 8. Düşmanın "tepki vermesi" için mesafe (px)

clock = pygame.time.Clock()
running = True # 9. Oyunu çalışır durumda tutun
```

BÖLÜM 2

KOD:

```
while running: # 10. Ana oyun döngüsü (her karede)
for event in pygame.event.get(): # 11. Tüm olayları kontrol
et
if event.type == pygame.QUIT: # 12. Pencere kapalıysa…
running = False # 13. …döngüyü durdur

# ---- HAREKET (while döngüsünün içinde olmalı) ----
keys = pygame.key.get_pressed() # 14. Hangi tuşlara basıldı?

Eğer tuşlar[pygame.K_LEFT] ise:
oyuncu.x -= hız # 15. Sola hareket et
Eğer tuşlar[pygame.K_RIGHT] ise:
oyuncu.x += hız # 16. Sağa hareket et
Eğer tuşlar[pygame.K_UP] ise:
oyuncu.y -= hız # 17. Yukarı hareket et
Eğer tuşlar[pygame.K_DOWN] ise:
oyuncu.y += hız # 18. Aşağı hareket et

# ---- BASİT YAPAY ZEKÂ: yakınsa rengi değiştir ----
px, py = oyuncu.merkez # 19. Oyuncu merkezi
ex, ey = düşman.merkez # 20. Düşman merkezi
dist = math.hypot(px - ex, py - ey) # 21. Mesafe

eğer dist < react_distance ise: # 22. 120 px'ten daha
yakınsa…
enemy_color = (255, 200, 0) # 23. …düşman döner sarı
aksi takdirde:
düşman_rengi = (220, 60, 60) # 24. Aksi takdirde, düşman
kırmızı kalır
```

BÖLÜM 3
KOD:

```
# ---- DRAW ----
screen.fill((240, 245, 255))# 25. Background
pygame.draw.rect(screen, (60, 120, 255), player)
# 26. Player (blue)
pygame.draw.rect(screen, enemy_color, enemy)
# 27. Enemy (reactive)
pygame.display.flip()  # 28. Show changes

clock.tick(60)# Limit to 60 FPS

pygame.quit()     # 29. Close safely
```

3. Adım – Bu Kodda Neler Oluyor?

Kavram	Explanation
Algı	Düşman, oyuncuya olan uzaklığını math.hypot fonksiyonunu kullanarak
Karar	Mesafe react_distance'tan küçükse, düşman tepki vermeye karar verir.
Eylem	Tepki olarak rengi değişir.

Bu “algıla → karar ver → harekete geç” döngüsü, çoğu oyun yapay zekâ sisteminin temelini oluşturur.
Bunun gibi basit tepkiler bile oyun dünyasının daha canlı hissettirmesini sağlar.

4. ADIM – DENEYIN, DEĞIŞTIRIN, GÖZLEMLEYIN

Kodun bu kısımlarını değiştirmeyi deneyin ve neler olduğunu gözlemleyin:

1. react_distance değerini 120'den 250'ye değiştirin - ne olur?
2. Düşmanın sarı yerine yeşil (0, 255, 0) olmasını sağlayın.
3. Oyuncunun hızını 8 artırarak daha hızlı hareket etmesini sağlayın.
4. Düşmanın yalnızca oyuncu yukarı veya aşağı hareket ettiğinde tepki vermesini sağlayabilir misiniz (örneğin, yalnızca y konumlarını karşılaştırarak)?

Öğrencileri programı çalıştırmadan önce tahminlerde bulunmaya teşvik edin

— ne olacağını düşünüyorlar?

5. Adım – Yansıma
Bu gerçekten “zeka” mı?

Neden veya neden değil?

Düşmanın davranışını nasıl daha zeki gösterebiliriz?

İpucu: Oyunlardaki gerçek yapay zeka aynı mantığı kullanır — saniyede birçok kez tekrarlanan basit kararlar, zeka yanılsaması yaratabilir.

KAYNAKLAR

Kitap, Wiki ve Dokümantasyon

- Python Yazılım Vakfı. Python 3 Dokümantasyonu. Şurada mevcuttur: https://docs.python.org
- Pygame Topluluğu. Pygame Dokümantasyonu ve Eğitimleri. Şurada mevcuttur: https://www.pygame.org/docs/ ve https://www.pygame.org/wiki/tutorials
- Python Arcade Projesi. Python Arcade Kütüphanesi Dokümantasyonu. Şurada mevcuttur: https://api.arcade.academy
- Real Python. Pygame: Python'da Oyun Programlamaya Giriş. Şurada mevcuttur: https://realpython.com/pygame-a-primer/
- Microsoft. Yeni Başlayanlar İçin Yapay Zeka Müfredatı.

  https://github.com/microsoft/AI-For-Beginners

ÖĞRENME PLATFORMLARINA BAĞLANTILAR

- Python Resmi Web Sitesi – Python programlamanın temellerini indirin ve öğrenin. (https://www.python.org/)
- Pygame Dokümantasyonu – Pygame'deki fonksiyonlar, olaylar ve grafikler için resmi kılavuz. (https://www.pygame.org/docs/)
- Arcade Kütüphanesi Dokümantasyonu – Python'da basit 2D oyunlar için alternatif bir kütüphane. (https://api.arcade.academy/en/latest/)
- Pygame Yapay Zeka Kılavuzu – Pygame için basit yapay zeka algoritmalarına örnekler. (https://pygame-ai.readthedocs.io/en/latest/guide.html)
- Real Python – Pygame ile Oyun Oluşturun – Yeni başlayanlar için dostane bir eğitim. (https://realpython.com/pygame-a-primer/)

- KidsCanCode Eğitimleri – Pygame öğrenmek için kolay projeler ve zorluklar. (https://kidscancode.org/blog/)

- Yeni Başlayanlar İçin Yapay Zeka (Microsoft) – Yapay zeka prensiplerini tanıtan açık kaynaklı eğitim içeriği. (https://microsoft.github.io/AI-For-Beginners/)

PRATİK ALIŞTIRMA

BENI YAKALAYABILIRSEN YAKALA

Amaç: Oyuncunun mavi bir kareyi hareket ettirerek nesneleri topladığı, kırmızı bir yapay zekâ düşmanının ise oyuncuyu takip ettiği küçük bir oyun oluşturmak. Bu alıştırma, öğrenilen tüm becerileri bir araya getiriyor: hareket, çarpışma tespiti ve basit bir yapay zekâ davranışı.

Adım 1 – Ortamınızı Hazırlayın
Pygame'in kurulu olduğundan emin olun:
pip install pygame

Ardından Python editörünüzü açın ve catch_me.py adında yeni bir dosya oluşturun.

Adım 2 – Oyun Fikrini Anlamak

Öğe	Tanım
Oyuncu	Ok tuşlarıyla hareket eder.
Hedef	"Rastgele ortaya çıkar; toplandığında puanı artırır."
Düşman (AI)	Yön ve mesafeyi hesaplayarak oyuncuyu takip eder.
Amaç	Düşman sizi yakalamadan önce mümkün olduğunca çok hedefi vurun.

Kod yazmadan önce düşünün: Bir düşman oyuncunun nerede olduğunu nasıl "bilebilir"?

3. ADIM – OYUNU ADIM ADIM OLUŞTURUN

BÖLÜM 1

KOD:

```
iimport pygame, random, math
pygame.init()

# --- Pencere ---
GENİŞLİK, YÜKSEKLİK = 800, 600
ekran = pygame.display.set_mode((GENİŞLİK, YÜKSEKLİK))
pygame.display.set_caption("Yakala Beni Eğer
Yakalayabilirsen")
saat = pygame.time.Clock()

# --- Renkler ---
MAVİ= (60, 120, 255)
KIRMIZI= (220, 60, 60)
YEŞİL = (60, 200, 60)
BEYAZ = (240, 245, 255)
SİYAH = (20, 20, 20)

# --- Varlıklar ---
oyuncu = pygame.Rect(100, 100, 40, 40)
# mavi kare (oyuncu)
düşman= pygame.Rect(600, 400, 40, 40)
# kırmızı kare (düşmanı kovalayan)
hedef = pygame.Rect(
random.randint(50, GENİŞLİK - 50),
# yeşil hedef: rastgele konum
random.randint(50, YÜKSEKLİK - 50),25,25)
```

3. ADIM – OYUNU ADIM ADIM OLUŞTURUN

BÖLÜM 2

KOD:

```
OYUNCU_HIZI = 5
DÜŞMAN_HIZI = 2.5
skor = 0

# Yazı Tipi (yedek yazı tipi sorunlarını önlemek için
varsayılan yazı tipini kullanır)
font = pygame.font.Font(None, 26)

# --- Oyun döngüsü ---
çalışıyor = True
çalışırken:

# 1) Olayları işleyin
pygame.event.get() içindeki her olay için:
eğer olay.type == pygame.QUIT ise:

çalışıyor = False

# 2) Oyuncu hareketi (ok tuşları)
tuşlar = pygame.key.get_pressed()
eğer tuşlar[pygame.K_LEFT] ise:
oyuncu.x -= OYUNCU_HIZI
eğer tuşlar[pygame.K_RIGHT] ise:
oyuncu.x += OYUNCU_HIZI
eğer tuşlar[pygame.K_UP] ise:
oyuncu.y -= OYUNCU_HIZI
eğer
```

3. ADIM – OYUNU ADIM ADIM OLUŞTURUN

BÖLÜM III

KOD:

```
#Oyuncuyu pencere sınırları içinde tutun
player.clamp_ip(screen.get_rect())

# 3) Basit Yapay Zeka: düşman oyuncuyu kovalıyor
dx = player.centerx - enemy.centerx
dy = player.centery - enemy.centery
dist = math.hypot(dx, dy)
if dist != 0: # ikisi çakıştığında sıfıra bölmeyi önleyin
enemy.x += int((dx / dist) * ENEMY_SPEED)
enemy.y += int((dy / dist) * ENEMY_SPEED)

# 4) Oyuncu hedefi topluyor
if player.colliderect(target):
score += 1
target.topleft = (
random.randint(25, WIDTH - 50),

random.randint(25, HEIGHT - 50)
)

# 5) Her şeyi çizin
screen.fill(WHITE)
pygame.draw.rect(screen, MAVİ, oyuncu)
pygame.draw.rect(ekran, KIRMIZI, düşman)
pygame.draw.rect(ekran, YEŞİL, hedef)
Send feedback)
```

3. ADIM – OYUNU ADIM ADIM OLUŞTURUN

BÖLÜM IV

KOD:

```
# Skoru Çiz
skor_metni = font.render(f"Skor: {skor}", True, SİYAH)
ekran.blit(skor_metni, (10, 10))

pygame.display.flip()
clock.tick(60)

pygame.quit()
```

4. Adım – Deney Yapın ve Düşünün

Bu değerleri değiştirmeyi deneyin ve neler olduğunu gözlemleyin:

Düşmanın hızını artırmak veya azaltmak için ENEMY_SPEED değerini değiştirin.
Oyuncunun veya hedefin rengini değiştirin.
Oyuncunun boyutunu küçülterek oyunu zorlaştırın.
Düşman oyuncuya dokunduğunda Oyun Bitti mesajı ekleyin.

DÜŞÜNME SORULARI

Bu modülde öğrendikleriniz hakkında derinlemesine düşünmek için zaman ayırın. Aşağıdaki soruları dikkatlice yanıtlayın:

- **Düşünme Sorusu 1:** Oyununuzu oluşturmanın en zor kısmı neydi?

- **Reflection Question 2:** How does the AI make the game feel more interactive?

- **Düşünme Sorusu 3:** Yapay zekanın insan eylemlerine tepki verdiği başka durumlar düşünebilir misiniz (örneğin, eğitim veya sağlık hizmetleri)?

- **Düşünme Sorusu 4:** Bu oyunu geliştirebilseydiniz, bir sonraki adımda ne tür bir yapay zeka davranışı eklerdiniz?

- **Düşünme Sorusu 5:** Düşmanı "zeki" gösteren nedir?

- **Düşünme Sorusu 6:** Mesafe hesaplaması (math.hypot) oyuncuyu bulmasına nasıl yardımcı oluyor?

- **Düşünme Sorusu 7:** Benzer bir mantık kullanan gerçek oyunlar düşünebilir misiniz?

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

1. Pygame'de oyun döngüsü ne yapar?
a) Oyun sırasında yeni kütüphaneler yükler
b) Oyun eylemlerini saniyede birçok kez tekrarlar
c) Oyun penceresini kapatır
d) Arka plan resmini yalnızca bir kez yükler

2. pygame.display.flip() fonksiyonunun amacı nedir?
a) Klavye girişini kontrol eder
b) Bir dikdörtgen çizer
c) Oyun penceresini yeni görsellerle günceller
d) Oyun döngüsünü duraklatır

3. Hangi fonksiyon hangi klavye tuşlarının basıldığını kontrol eder?
a) pygame.quit()
b) pygame.key.get_pressed()
c) pygame.event.get()
d) pygame.display.set_mode()

4. Yapay zeka örneğinde, düşmanın oyuncuya tepki vermesini sağlayan nedir?
a) Zamanlayıcı olayı
b) Rastgele hareket
c) Oyuncu ile düşman arasındaki mesafe
d) Oyuncunun rengi

5. Kodda react_distance değerini artırırsanız ne olur?
a) Düşman daha uzaktan tepki verir
b) Düşman daha hızlı hareket eder
c) Oyuncu yavaşlar
d) Pencere boyutu değişir

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

6. Yapay zeka kodundaki `math.hypot()` fonksiyonunun amacı nedir?
a) İki nokta arasındaki mesafeyi hesaplamak
b) Düşmanı ekrana çizmek
c) Yeni bir rastgele konum oluşturmak
d) Duvarlarla çarpışmaları tespit etmek

7. Bu kod satırı ne yapar?
a) Ekrana bir metin çizer
b) Oyuncuyu temsil etmek için bir dikdörtgen oluşturur
c) Yapay zeka algoritmasını başlatır
d) Arka plan rengini değiştirir

8. Döngüden `pygame.display.flip()`'i kaldırırsanız ne olur?
a) Program daha hızlı çalışır
b) Pencere güncellenmez ve donmuş görünür
c) Renkler otomatik olarak değişir
d) Oyuncu iki kat daha hızlı hareket eder

8. Aşağıdakilerden hangisi bir oyunda yapay zeka davranışına örnektir?
a) Ekrana bir kare çizmek
b) Bir karakterin yalnızca oyuncu bir tuşa bastığında hareket etmesini sağlamak
c) Bir düşmanın oyuncuyu takip etmesini veya ondan kaçınmasını sağlamak
d) Arka plan müziği yüklemek

8. Yapay zeka oyunları nasıl daha ilginç hale getirebilir?
a) Hataları otomatik olarak düzelterek
b) Dinamik zorluklar ve duyarlı karakterler oluşturarak
c) Grafikleri otomatik olarak değiştirerek
d) Klavyeyi kontrol ederek

Modül 5: Roboflow ile Nesne Tanıma – Bilgisayarlı Görüye Giriş

GİRİŞ

Bu modülün amacı, bilgisayar görüşüne kapsamlı bir giriş sağlamak, makinelerin görsel verileri nasıl algıladığını ve işlediğini açıklamak ve öğrencileri Roboflow platformunu ve YOLO gibi gelişmiş algoritmaları kullanarak kendi nesne tanıma modellerini geliştirmeleri konusunda yönlendirmektir.

Modülün sonunda, aşağıdaki gibi farklı beceriler kazanabileceksiniz:

- **Temel Kavramları Anlamak:** Görüntü işleme ve bilgisayar görüşü arasındaki farkı ayırt edin ve bilgisayarların görüntüleri veri (piksel ve ikili kod) olarak nasıl yorumladığının mantığını anlayın.
- **Bilgisayarlı Görü Sürecinin Uygulanması:** Bilgisayar görüşü sürecinin beş temel adımını uygulayın: veri elde etme, veri hazırlama, yapay zeka modelini belirleme, modeli eğitme ve sonuçları yorumlama.
- **Roboflow Kullanımı:** Projeler oluşturmak, veri kümeleri yüklemek ve görüntü verilerini etkili bir şekilde yönetmek için Roboflow platformunda gezinmeyi öğrenin.
- **Veri Etiketleme:** Yapay zeka modeli eğitimi için veri kümelerini hazırlamak amacıyla doğru veri etiketlemesi (etiketleme) gerçekleştirin.
- **Model Eğitimi**: Belirli nesneleri tanımlamak için YOLOv11 gibi gelişmiş nesne algılama modellerini seçin ve eğitin.

Modül Süresi	2 saat (1 saat öğrenme + 1 saat uygulamalı egzersizler)

EĞİTİM MATERYALLERİ

ÜNİTE 5.1 BİLGİSAYARLI GÖRÜ İŞLEME NEDİR?

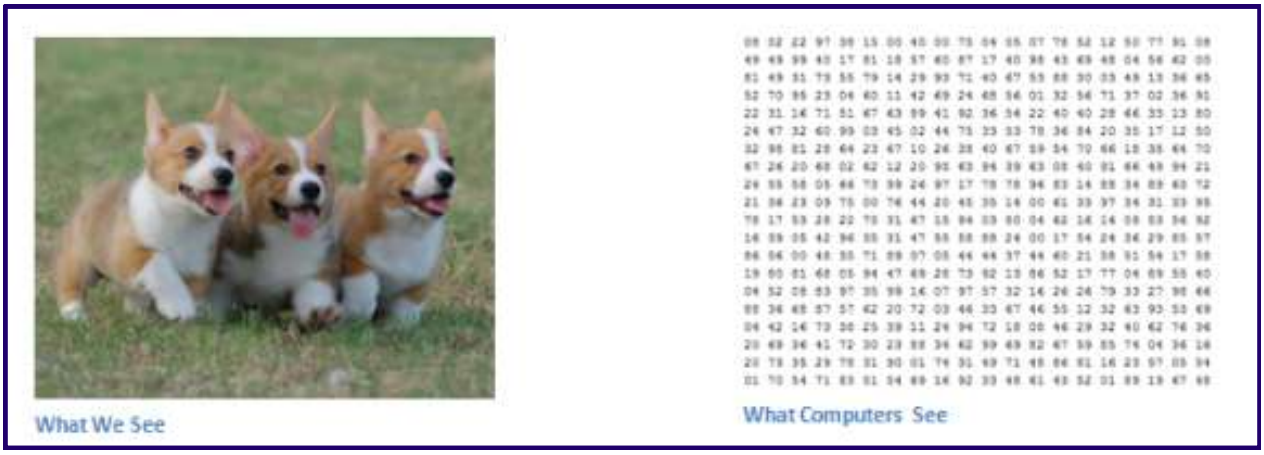
Görüntü işleme, dijital bir görüntüde bulunan önemli bilgileri tespit etmek, çıkarmak ve değerlendirmek için kullanılan bir tekniktir. Bu yöntem, insan görsel algısına benzer şekilde, bir görüntüdeki nesneler veya ortamlar hakkında anlamlı veriler üretir. Görüntü işlemenin temel amacı, görüntülerden anlam çıkarmak ve bu bilgiyi çeşitli alanlarda kullanılmak üzere sunmaktır. Görsel bilgi işleme teknolojisi uzun zamandır var olmasına rağmen, bu süreç geçmişte büyük ölçüde insan müdahalesine dayanıyordu. Bu hem zaman alıcıydı hem de hatalara açıktı. Örneğin, önceki yüz tanıma sistemlerinde, geliştiricilerin binlerce fotoğrafı manuel olarak etiketlemesi gerekiyordu; burun genişliği veya göz mesafesi gibi belirli özellikler tek tek tanımlanıyordu. Bunun nedeni, görüntü verilerinin genellikle düzensiz ve bilgisayarların anlaması zor olmasıydı. Bu nedenle, süreci otomatikleştirmek güçlü işlemciler ve gelişmiş bilgi işlem teknolojileri gerektiriyordu.

Günümüzde, gelişen işlem gücünün yardımıyla, bilgisayar görüşü uygulamaları, nesne tanımlama ve yüz tanıma, ayrıca sınıflandırma, öneri, izleme ve algılama için bu verileri doğru bir şekilde işlemek üzere yapay zeka ve makine öğrenimini (YZ/ML) kullanmaktadır. Bilgisayar görüşü, insan görsel sisteminden ve beyinle olan ilişkisinden esinlenerek, basitçe "bu görevi makinelerin yapmasına izin vermek" olarak tanımlanabilir. Bilgisayarlar gibi makinelerin görüntüleri ve videoları algılaması, analiz etmesi ve makine öğrenimi ve yapay zeka kullanarak bunları anlamlı hale dönüştürme sürecidir. 1950'lerde ortaya çıkan bilgisayar görüşü, 1970'lerde daktilo ve el yazısı arasındaki farkı tespit etme uygulamasıyla ticarileşmeye başladı. Bugün, kişi algılama ve tanıma teknolojileri, ürün kalite kontrol sistemleri, otonom araçlar, tarımsal uygulamalar, sağlık, eğitim ve savunma dahil olmak üzere birçok alanda kullanılmaktadır.

Görüntü işleme, fotoğraflar ve videolar da dahil olmak üzere yakalanan görüntü verilerinin işlenmesini, renk ve parlaklığının geri yüklenmesini ve ayarlanmasını içerir. Bilgisayar görüşü ise, görüntü işleme tekniklerinin görüntülere uygulanması ve bu sürecin yapay zeka ile birleştirilmesidir. Örneğin, görüntü işleme bir görüntü içindeki bir nesnenin kenarlarını bulmak için kullanılırken, bu nesneyi tanımlamak ve sınıflandırmak için makine öğrenmesi tekniklerinin kullanılması bilgisayar görüşü alanına girer.

ÜNİTE 5.2 BİLGİSAYAR NASIL GÖRÜR?

Dünya genelindeki neredeyse tüm bilgisayarların yapı taşları olan "hücreler", 1 ve 0 rakamlarından oluşur. Bu, bilgisayarların kullanıcıların iletmek istedikleri verileri kendi dillerinde, yani 1'ler ve 0'lar olarak algıladığı anlamına gelir. Sonuçta, birçok görüntüden oluşan resimler ve videolar da veridir, bu nedenle bilgisayarlar bunları sıfırlar ve birler veya ikili sayı sistemi olarak algılar. Bunu daha iyi açıklamak için aşağıdaki resmi kullanabiliriz.



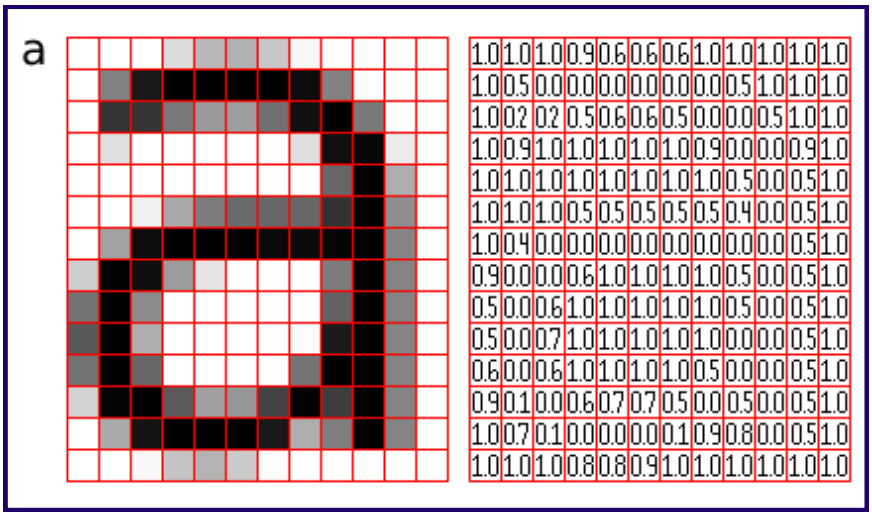
FŞekil 7 Görünüm Biçimi: Video gözetimi için sinir ağlarının kullanımı (2025) https://www.videonet9.com/using-neural-networks-for-video-surveillance.html adresinden; Ekleme Tarihi: 09.08.2025.

Ama eğer bu resimde iki basamaklı değerler gösteriliyor ama sıfır veya bir yok diye merak ediyorsanız, evet, bu doğru. Bilgisayar işlemcilerimiz bu değerleri sıfır ve birlere dönüştürerek yorumluyor. Eğer bu değerler de sıfır ve bir olarak gösterilseydi, çok uzun ve çirkin bir görüntü olurdu.

Aslında, yukarıdaki resimdeki bu iki basamaklı değerlerin her birine piksel denebilir. Peki, piksel nedir? Dijital ekranlarda görüntünün oluşturulmasını ve kontrol edilmesini sağlayan en küçük birimdir.

ÜNİTE 5.3 ŞİMDİ BİR ALIŞTIRMA YAPALIM

- Kareli bir defter alın.
- Kareli defterin her karesini 0 ile 1 arasında sayılarla numaralandırın.
- Herhangi bir kareye yazılan sayı 1'e ne kadar yakınsa, kare o kadar beyaz olur.
- Sayı 0'a ne kadar yakınsa, kare o kadar siyah olur.
- Sayı 0,5 ise, kareyi gri renkle boyayın



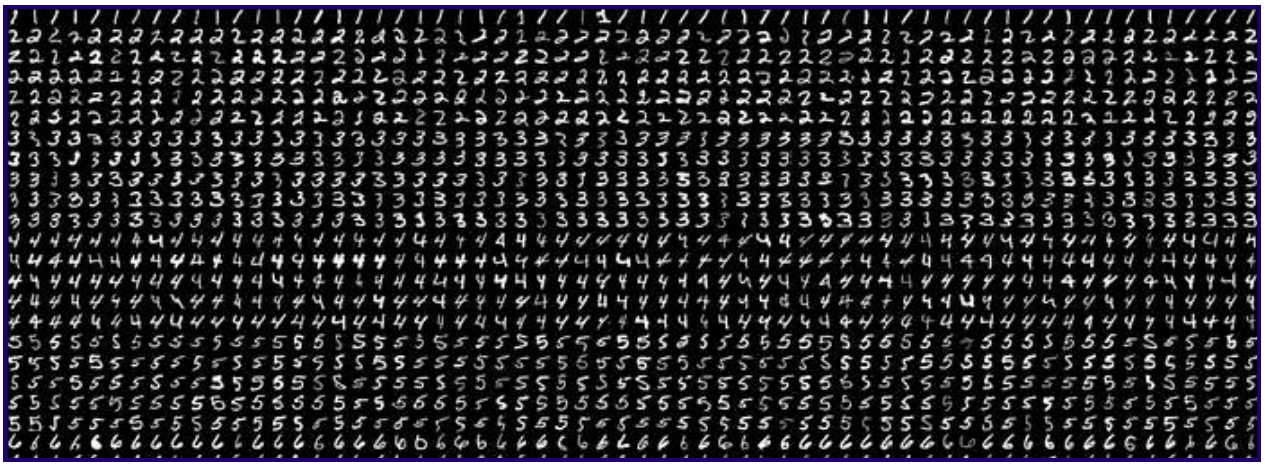
Şekil 8 Bilgisayarın pikseli nasıl gördüğü. (2025) https://medium.com/@meriyananjalika99/from-pixels-to-predictions-getting-started-with-cnns-convolutional-neural-networks-e3f84781cc1e adresinden; Ekleme Tarihi: 09.08.2025.

Bu tür bir deftere dikkatlice çizim yaparsanız, uzaktan bakıldığında normal bir çizim gibi görünecektir. Yukarıdaki resim için bunu deneyin. Ekrandan birkaç metre uzaklaşın. Resim daha normal görünecektir. Aslında, bilgisayar ve televizyon ekranları görüntüleri ve videoları bu şekilde milyonlarca küçük kare kullanarak gösterir. Görüntüleri kaydetmek için, bu kareli defterdeki sayıları bellekte saklarlar. Pikselli dediğimiz düşük kaliteli görüntülerde, bu kareli defterdeki tek tek kareleri görebilirsiniz.

ÜNİTE 5.4 BILGISAYARLA GÖRME SÜREÇLERI TEMEL OLARAK BEŞ ADIMDAN OLUŞUR:

1. Görüntü/veri elde etme
2. Görüntü/verilerin hazırlanması
3. Kullanılacak yapay zeka modelinin belirlenmesi
4. Kullanılacak yapay zeka modelinin eğitilmesi
5. Sonuçların yorumlanması

Aşağıda basit bir el yazısı rakam tanıma uygulaması için elde edilen görüntüler gösterilmektedir. Bu görüntüler bilgisayar dostu gri tonlamaya dönüştürülmüş ve yapay zekayı eğitmek için kullanılmıştır. Test sonuçları, eğitimden sonra %99 doğruluk oranı göstermektedir.

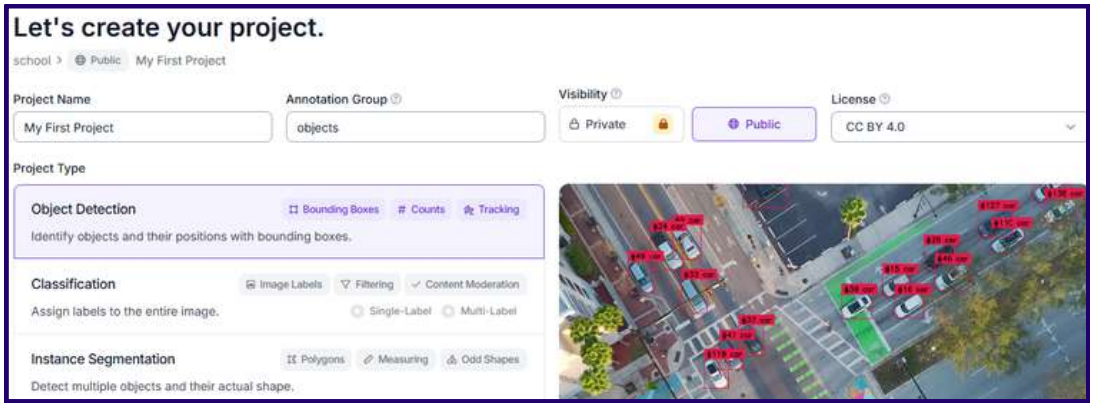


FŞekil 9 Kaggle Veri Kümesi (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.

Roboflow, bilgisayar görüşü projeleri geliştirmek için tasarlanmış kapsamlı bir yapay zeka platformudur. Görüntü ve video verilerini kullanarak bilgisayar görüşü modelleri oluşturma, eğitme ve dağıtma sürecini basitleştirir. Hem yeni başlayanlar hem de deneyimli geliştiriciler için kullanımı kolay bir arayüz sunar.

ÜNİTE 5.5 ROBOFLOW'A NASIL KAYIT OLUNUR?

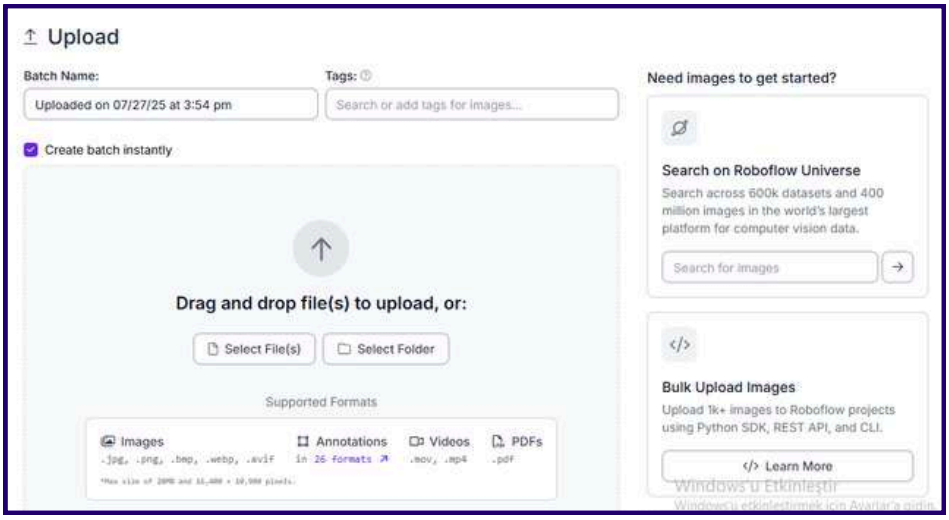
- https://roboflow.com/ adresine gidin.
- Sağ üst köşedeki Oturum Aç düğmesine tıklayın.
- Açılan pencerede, varsa üyelik bilgilerinizi girin ve sisteme giriş yapın.
- İlk kez giriş yapanlar için, Google hesabı ile devam etmek en pratik yöntem olduğu için Google ile Devam Et seçeneğini kullanın.
- Roboflow sitesinde kullanmak istediğiniz Google hesabını seçtikten ve gerekli izinleri verdikten sonra oturum açmış olacaksınız.
- Açılan sayfada YENİ PROJE düğmesine tıklayın, proje bilgilerini girin ve projeyi oluşturun.



Şekil 10 Kaggle Veri Seti (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.

Açılan yeni sayfada, yapay zekamız için eğitim, doğrulama ve test verilerini ekliyoruz. Bundan sonra bu verilere "veri seti" diyeceğiz. Eğer bir veri setimiz yoksa, RoboFlow Universe sistemi içindeki hazır veri setlerini kullanabiliriz. (https://universe.roboflow.com)

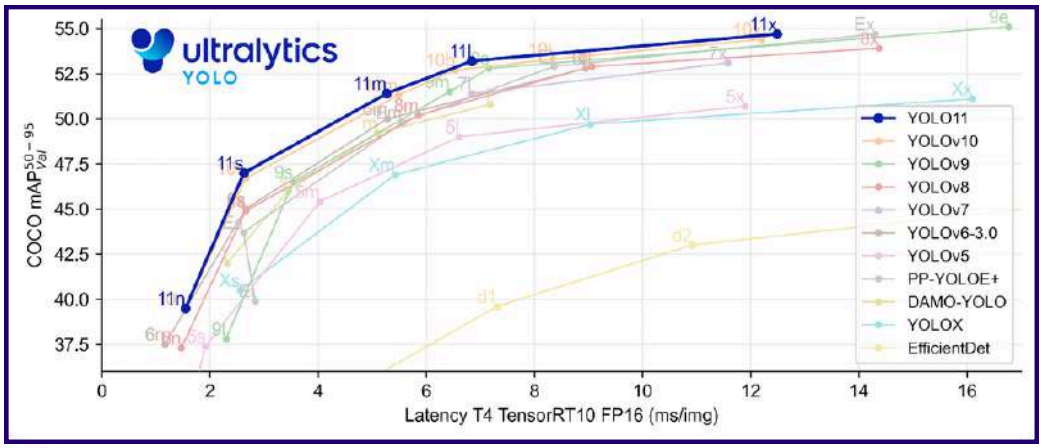
Benzer şekilde, Kaggle da hazır veri setleri sunan bir web sitesidir. (https://www.kaggle.com/)



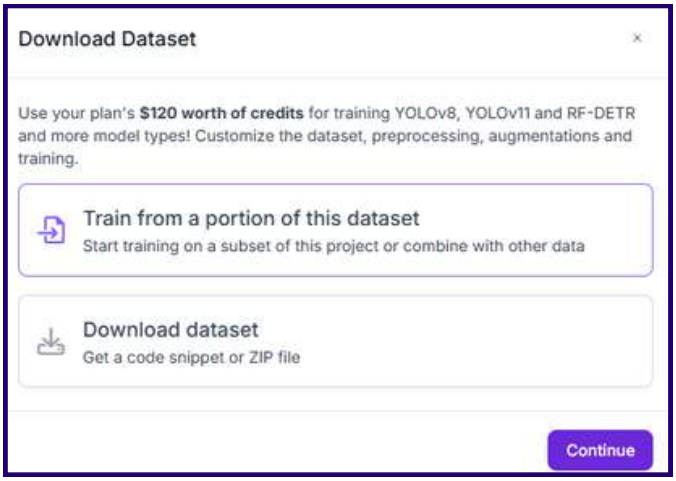
Şekil 11 Roboflow (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.

Bu sistemde, önceden hazırlanmış bir veri seti kullanarak işlem adımlarına devam edeceğiz. Bu nedenle, RoboFlow üzerinden https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw adresindeki veri setini kullanacağız.

Bu adresi girdikten sonra, veri setini indir düğmesine tıklayın. Ancak, bu düğmeye tıklamadan önce, hangi yapay zeka modelini kullanacağımıza karar vermemiz gerekiyor. Bu uygulamada, nesne algılamada daha verimli olan YOLOv11 yapay zeka modelini kullanacağız. İlgili algoritmanın GitHub profilini (https://github.com/ultralytics/ultralytics) ziyaret ederseniz, farklı önceden hazırlanmış ağırlıkları ve ayrıntılı bilgileri bulabilirsiniz. (Yolov3 çevrimiçi testi https://v-iashin.github.io/detector)

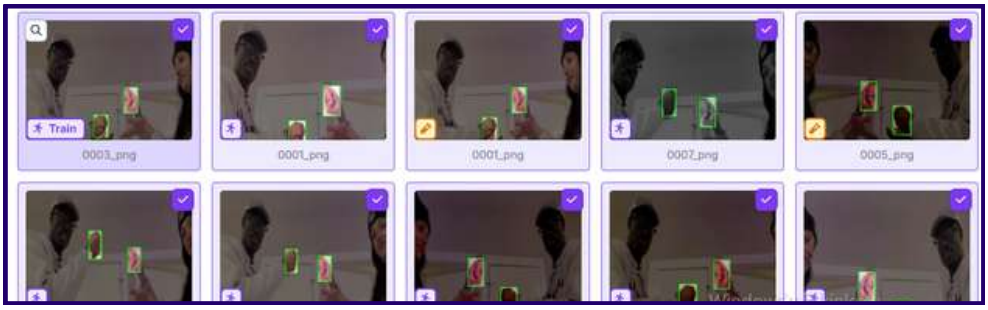


Şekil 12 Veri kümesi (2025) www.roboflow.com adresinden alınmıştır; Ekleme Tarihi: 09.08.2025.

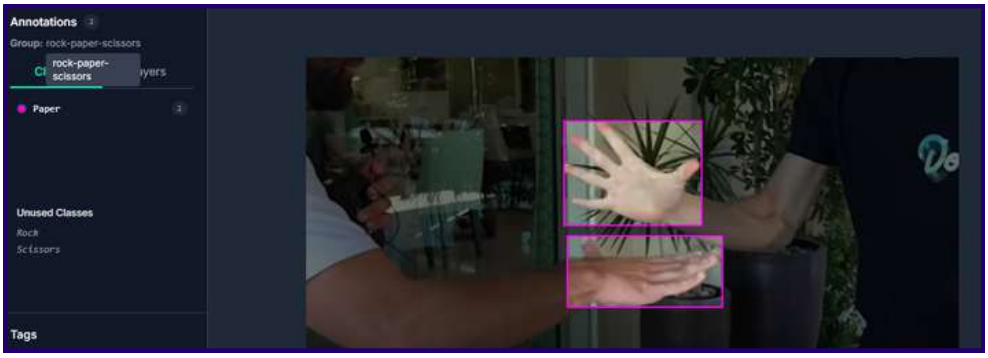


Şekil 13 Veri Kümesi (2025) www.roboflow.com adresinden alınmıştır; Ekleme Tarihi: 09.08.2025.

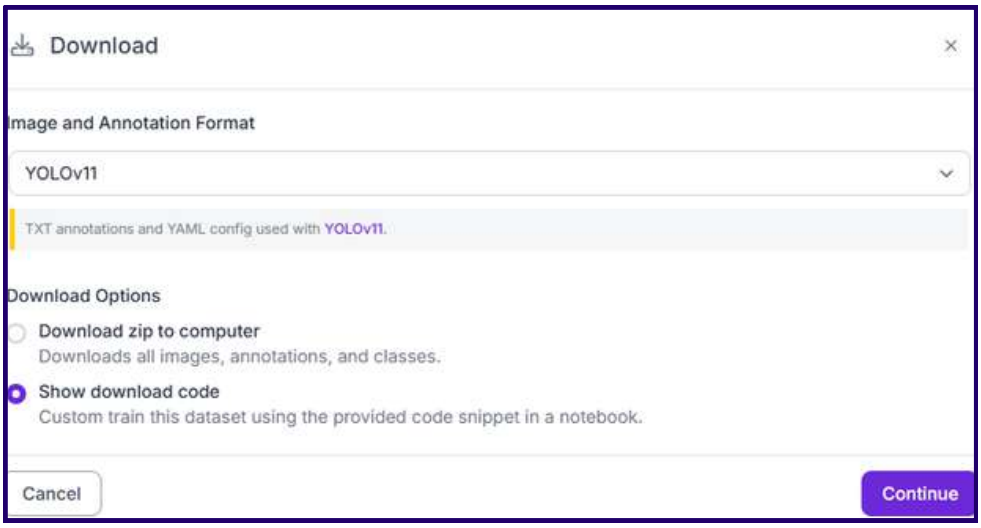
Veri setindeki verileri etiketleyip kontrol ettikten sonra, "Veri setini indir" düğmesine tıklayın, modeli ve çevrimiçi yöntemi seçin ve site tarafından sağlanan Python kodunu alın.



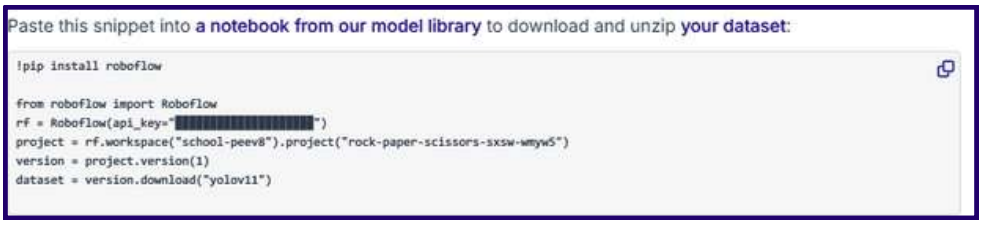
Şekil 13 Veri Kümesi (2025) www.roboflow.com adresinden alınmıştır; Ekleme Tarihi: 09.08.2025.



Şekil 15 Kesme Noktası (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.



Şekil 16 YOLOv11 (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.



Şekil 17 Alıntı (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025.

Bir sonraki adımda, Roboflow tarafından bizim için hazırlanan ve Google Colab'da çalışan YOLOv11 algoritmasının kodunu açacağız. (https://blog.roboflow.com/yolov11-how-to-train-custom-data/)

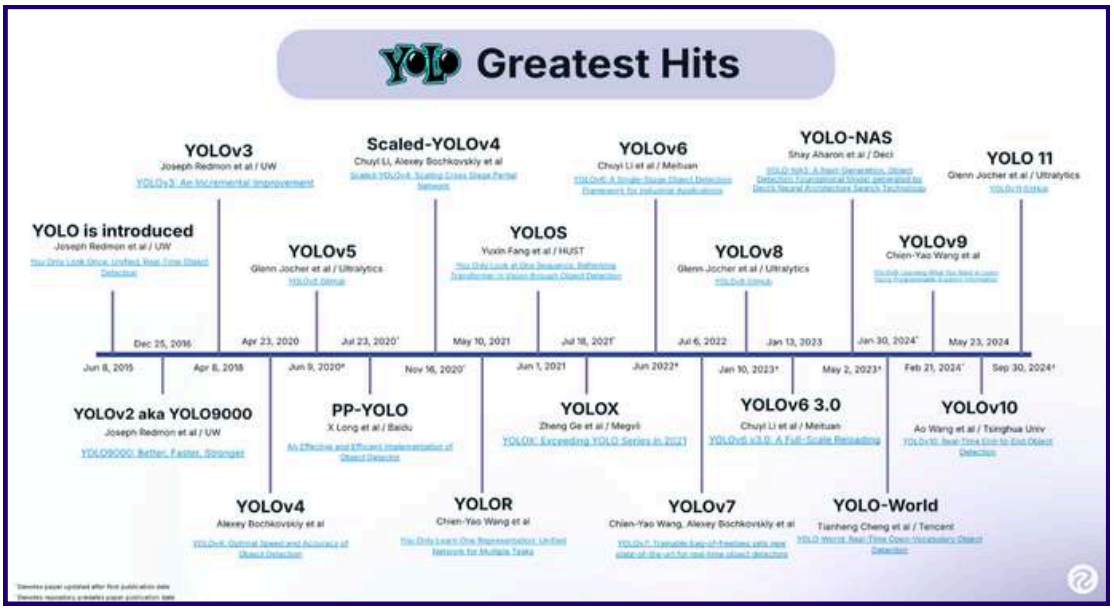
https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolo11-object-detection-on-custom-dataset.ipynb?ref=blog.roboflow.com adresini ziyaret edin ve oradaki talimatları izleyin.Başka bir seçenek ise bir model oluşturup Roboflow'da kullanmaktır. Bu daha hızlı ve kolay olabilir, ancak YOLOv11 daha iyi ve daha doğru sonuçlar sağlar.

Son birkaç yıldır, iki nesne algılama mimarisi bilgisayar görüşü dünyasını kasıp kavurdu: Evrişimsel Sinir Ağları (CNN'ler) ve Yalnızca Bir Kez Bak (YOLO). Hem CNN'ler hem de YOLO, modern bilgisayar görüşü endüstrisinin şekillenmesine yardımcı oldu. Transformer'lar ayrıca bilgisayarla görme ve nesne algılama alanlarında da giderek daha önemli bir rol oynamaktadır.
Geri dönüt gönderin

CNN'ler, bir görüntü hakkında özellikler öğrenmek için bir görüntü işleme tekniği olan evrişimleri kullanır. Bu işlem, özellikleri öğrenmek için görüntüdeki her piksele kayan bir pencere uygulanmasını içerir. Evrişimlerden elde edilen bilgiler bir sinir ağı tarafından işlenir. R-CNN, Mask R-CNN ve Fast R-CNN dahil olmak üzere birçok CNN uygulaması mevcuttur.

YOLO model ailesi de bilgisayar görüşü dünyasında etkili olmuştur. 2014 yılında Joseph Redmon tarafından tanıtılan YOLO, aktif bir araştırma ve geliştirme alanı haline gelmiş, geniş bir topluluk tarafından benimsenmiş ve birçok geliştirici ve araştırmacı tarafından uygulanmıştır. Ultralytics ekibi tarafından geliştirilen ve iyileştirilen YOLOv5 ve YOLOv8, dünya çapında nesne algılama modellerinin üretimini desteklemektedir.

YOLO hakkında daha fazla bilgi şu adreste bulunabilir:
https://blog.roboflow.com/guide-to-yolo-models



Şekil 18 YOLO-Tarihçesi (2025) www.roboflow.com adresinden; Ekleme Tarihi: 09.08.2025

Dünya çapında bilgisayar görüşü alanında aktif olarak kullanılan algoritmaların ayrıntılı sıralamasını analiz etmek için ilgili adresi ziyaret edebilirsiniz:
https://leaderboard.roboflow.com/?ref=blog.roboflow.com

KAYNAKLAR

Kitaplar ve Belgeler

- Banerjee, Chayan & Fookes, Clinton & Karniadakis, George. (2023). Fizik Bilgili Bilgisayar Görüşü: Bir İnceleme ve Perspektifler. 10.48550/arXiv.2305.18035.
- LeCun, Y. (2025). MNIST Veri Kümesinin Uzaktan Görünümü http://yann.lecun.com/exdb/mnist/. Tarih Ekleme: 09.08.2025.
- Mohite, Amruta & Kulkarni, Atharva & Chitnis, Rutwik & Mane, Swapnil & Asabe, Shubham. (2021). Yapay Zeka Denetimi: Görsel Denetim İçin Bilgisayar Görüşü. Uluslararası İleri Bilgisayar Bilimi ve Yönetimi Araştırma Dergisi. 7. 29.
- Nesne Tanıma (2025). https://viso.ai/product/computer-vision-parking-lot-occupancy-tutorial/ adresinden alınmıştır. Eklenme tarihi: 10.08.2025.
- Video gözetimi için sinir ağlarının kullanımı (2025) https://www.videonet9.com/using-neural-networks-for-video-surveillance.html adresinden alınmıştır. Eklenme tarihi: 09.08.2025.

ÖĞRENME PLATFORMLARINA BAĞLANTILAR

Dünya genelinde bilgisayar görüşü alanında aktif olarak kullanılan algoritmaların detaylı sıralamasını analiz etmek için ilgili adresi ziyaret edebilirsiniz: https://leaderboard.roboflow.com/?ref=blog.roboflow.com

Roboflow, bilgisayar görüşü projeleri geliştirmek için tasarlanmış kapsamlı bir yapay zeka platformudur. https://roboflow.com/.

PRATİK ALIŞTIRMA

NESNE AVCILARI – ÖĞRETMEN KILAVUZU

***Etkinlik Amacı**
*Bilgisayar görüşü mantığını anlamak
*Roboflow kullanarak nesne tanıma modeli oluşturmak
*Veri toplama, etiketleme, model eğitimi ve test süreçlerini deneyimlemek

***Hazırlık Listesi**
*İnternet bağlantılı bilgisayarlar veya tabletler
*4-5 kişilik öğrenci grupları
*Akıllı telefonlar (fotoğraf çekmek için)
*Ücretsiz Roboflow hesabı
*Basit nesneler (kitap, kalem, su şişesi vb.)

Süre ve Program*
*Bölüm 1: Bilgisayar Görüşüne Giriş – 15 dakika
*Bölüm 2: Fotoğraf Toplama – 20 dakika
*Bölüm 3: Etiketleme – 25 dakika
*Bölüm 4: Model Eğitimi – 20 dakika
*Bölüm 5: Test ve Yarışma – 20 dakika

Uygulama Adımları

1. Öğrencilere kısa bir sunumla bilgisayarla görme kavramını tanıtın.
2. Gruplar iki farklı nesne seçer ve farklı açılardan 15-20 fotoğraf çeker.
3. Roboflow'a giriş yapın ve yeni bir proje oluşturun.
4. Fotoğraflar yüklenir ve nesneler çerçeve çizilerek etiketlenir.
5. Model eğitildi ve test edildi.
6. Gruplar birbirlerinin modellerini test eder ve puanlar verilir.

NESNE AVCILARI – ÖĞRETMEN KILAVUZU

Oyunlaştırma Fikirleri
Zaman Yarışı: Kim en hızlı veri toplayıp etiketleyebilir?
Karma Test: Model, daha önce hiç görmediği nesnelerle test edilir.
Hata Avı: Modelin yanlış olduğu durumlar bulunur ve tartışılır.

NESNE AVCILARI – ÖĞRENCI ETKINLIK SAYFASI

Bilgisayar Görüşü Nedir?
Bilgisayar görüşü, bilgisayarların kameralar veya görüntüler kullanarak nesneleri algılamasına ve anlamasına olanak tanıyan yapay zeka alanıdır.

Görevleriniz

1. Takım olarak iki nesne seçin.
2. Her nesnenin 15-20 fotoğrafını farklı açılardan çekin.
3. Roboflow'da bir proje açın ve fotoğrafları yükleyin.
4. Nesneleri etiketleyin.
5. Modelinizi eğitin.
6. Diğer takımların nesnelerini tanımaya çalışın.

Fotoğraf Çekme İpuçları
Farklı açılardan çekim yapın.
Farklı ışık koşullarını deneyin.
Nesneyi hem yakından hem de uzaktan çekin.

Etiketleme Kılavuzu
Fotoğraftaki nesneyi kare içine alın ve adını yazın.
Örnek: Kurşun kalem, Silgi.

Gözlemlerim

Modelimin en iyi tanıdığı durumlar
- Modelimin zorlandığı durumlar

DÜŞÜNME SORULARI

Bu modülde öğrendikleriniz hakkında derinlemesine düşünmek için zaman ayırın. Aşağıdaki soruları dikkatlice yanıtlayın:

- **Piksellerden Algıya:** Bu modülde, bilgisayarların görüntüleri şekiller ve renkler yerine sayılar ve ikili kodlardan oluşan ızgaralar olarak algıladığını öğrendiniz. Dünyaya bu "sayısal" bakış açısı, bir yapay zekanın kalem ve silgi gibi iki farklı nesneyi nasıl ayırt ettiğine dair anlayışınızı nasıl değiştiriyor?

- **Veri Çeşitliliğinin Önemi:** "Nesne Avcıları" etkinliği, farklı açılardan ve çeşitli aydınlatma koşulları altında fotoğraf çekmeyi vurguladı. Yapay zekanın gerçek dünya ortamında bir nesneyi doğru bir şekilde tanıması için eğitim veri setinizdeki bu çeşitlilik neden çok önemlidir?

- **Doğruluk ve "Hataları" Analiz Etme:** Test aşamasında, modelin yanlış olduğu durumları belirlemeniz teşvik edildi - bir "Hata Avı". Gözlemlerinize dayanarak, hangi belirli çevresel faktörler (gölgeler veya mesafe gibi) modelinizin doğru çalışmasını zorlaştırdı?

- **Gerçek Dünya Problemlerinin Çözümü:** Bilgisayar görüşü halihazırda sağlık, otonom araçlar ve tarım gibi alanlarda kullanılmaktadır. Kendi modelinizi eğittikten sonra, okulunuzda veya yerel topluluğunuzda bu teknoloji kullanılarak çözülebileceğine inandığınız belirli bir problem nedir?

ÇOKTAN SEÇMELİ SORULARDAN OLUŞAN SINAV (her soru için bir doğru cevap)

1. Bilgisayar görüşünün birincil amacı nedir?
a) Bilgisayarların insan beyni gibi düşünmesini sağlamak.
b) Makinelerin dünyayı insan gözü gibi "görmesini" ve yorumlamasını sağlamak.
c) Büyük veri kümelerini veritabanlarında depolamak.
d) Sadece metin tabanlı verileri analiz etmek.

2. Aşağıdakilerden hangisi bilgisayar görüşünün tipik bir uygulaması DEĞİLDİR?

a) Otonom araçlarda yol işaretlerini tanımak.
b) Yüz tanıma sistemleri.
c) Ürün öneri sistemlerinde kullanıcı tercihlerini analiz etmek.
d) Tıbbi görüntülerde tümörleri tespit etmek.

3. Bir görüntüdeki belirli nesnelerin konumunu ve türünü belirlemeyi amaçlayan bilgisayar görüşü görevi nedir?

a) Görüntü Sınıflandırması
b) Nesne Tespiti
c) Görüntü Bölümlemesi
d) Özellik Çıkarma

4. Aşağıdakilerden hangisi bilgisayar görüşü sistemlerinin gerçek dünyada karşılaştığı zorluklardan biridir?

a) Veri kıtlığı ve bu verilerin çeşitliliği.
b) Görsel verilerin düşük boyutluluğu ve basit yapısı.
c) Aydınlatma, konum ve ölçekte tutarlılık.
d) Modellerin sürekli olarak mükemmel doğrulukta sonuçlar elde etmesi

5. Aşağıdaki ifadelerden hangisi bir modelin aşırı uyum sağladığını gösterir?

a) Hem eğitim hem de test verilerinde düşük doğruluk.
b) Eğitim verilerinde çok yüksek doğruluk, test verilerinde düşük doğruluk.
c) Hem eğitim hem de test verilerinde yüksek doğruluk.
d) Model her görüntüyü aynı şekilde etiketliyor.

Özet ve Sonraki Adımlar

İlk derste oluşturulan temel bilgiler üzerine inşa edilen bu ileri düzey öğrenme materyalleri, ortaokul öğrencilerini yapay zekanın pratik uygulamalarına daldırmak için tasarlanmıştır. İçerik, teori ve pratik arasındaki boşluğu doldurarak, öğrencileri Python ve Scratch kullanarak kolay ve temel yapay zeka modelleri oluşturma konusunda yönlendirir. "Yapay Zeka Kaçış Odası"nda karar vermenin ardındaki matematiksel mantığı keşfetmekten, oyun karakterleri için makine öğrenimi modelleri eğitmeye ve Roboflow ile bilgisayar görüşü uygulamaları geliştirmeye kadar, modüller kapsamlı ve uygulamalı bir deneyim sunar. Aylin ve Alara gibi ilişkilendirilebilir karakterler içeren anlatı odaklı bir yaklaşım, sinir ağları, K-ortalama kümeleme ve nesne tanıma gibi karmaşık teknik kavramların erişilebilir ve ilgi çekici kalmasını sağlar. Bu kursu tamamlayarak, öğrenciler pasif tüketicilerden aktif teknoloji yaratıcılarına dönüşeceklerdir. Temel akıllı sistemleri kodlamak için teknik yeterlilik, model doğruluğunu değerlendirmek için eleştirel düşünme becerileri ve yapay zekayı sorumlu bir şekilde kullanmak için etik farkındalık kazanacaklardır. Programlamaya ve veri bilimine yönelik bu kapsamlı giriş, öğrencilerin STEM alanındaki yetkinliklerini pekiştirmenin yanı sıra, teknoloji odaklı bir dünyada gelecekteki akademik ve profesyonel hedeflerine de hazırlıyor.

Geri bildirim gönder

Sırada Ne Var?

Bu ileri düzey konuların sınıfa sürdürülebilir bir şekilde entegre edilmesini desteklemek amacıyla, proje bir sonraki aşamada "Okul Eğitimcileri için E-Araç Seti"ni tanıtacaktır. Bu kaynak, öğretmenlere yapay zeka eğitimini etkili bir şekilde sunmak için ayrıntılı ders planları ve metodolojik rehberlik sağlayacaktır. Ayrıca, öğrencilerin yeni edindikleri becerilerini sergilemeleri için yaklaşan "Yapay Zeka Gelecek Kaşifleri" yarışmalarına katılmaları teşvik edilmektedir; bu yarışmalarda öğrenciler bilgilerini gerçek dünya sorunlarını çözmek için uygulayabilir ve daha geniş bir genç yenilikçi topluluğuyla bağlantı kurabilirler.

UYGULAMALI PYTHON VE SCRATCH EĞITIM MATERYALLERI

Proje Ortakları

FUTURE-STEM-HUB

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

KÜTAHYA
İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

tetra solutions

M&M profuture training

ETAP ESCOLA PROFISSIONAL

Project Coordinator:
University of Duisburg-Essen, Germany

Email Address
mustafa.bilgin@uni-due.de

Website
www.future-stem-hub.eu

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.