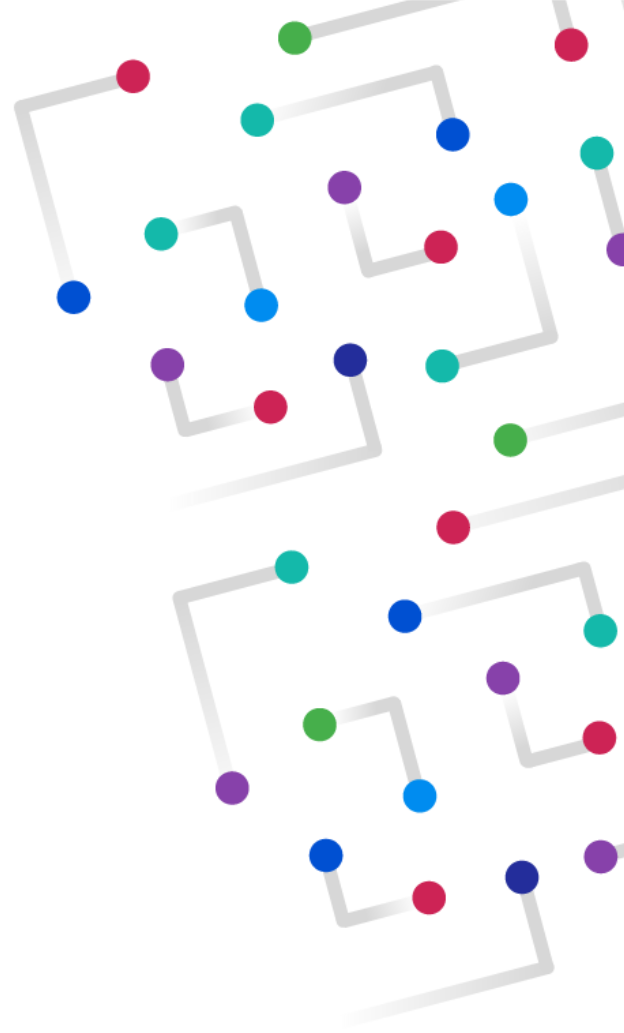


Raspberry Pi
Foundation

Experience AI

Słownik
terminów AI



Experience AI – słownik terminów

W słowniku wyjaśniono kluczowe terminy związane ze sztuczną inteligencją (artificial intelligence, AI) i uczeniem maszynowym (machine learning, ML) używane na lekcjach [Experience AI](#) i w innych materiałach.

[Przejdź do słownika](#)

Poniższe wyjaśnienia zostały przez nas opracowane przede wszystkim dla nauczycieli i edukatorów, z myślą o młodych odbiorcach. Słownik ten ma ułatwić odbiorcom zrozumienie kluczowych terminów i pogłębienie wiedzy technicznej.

Słownictwo jest istotnym elementem nauczania i uczenia się. Poprawne słownictwo wspiera uczniów w rozwoju ich sposobu rozumienia, natomiast stosowanie go w sposób niespójny może prowadzić do powstawania alternatywnych pojęć (błędnych przekonań), które zaburzają proces nauki. Więcej na temat alternatywnych pojęć przeczytasz w [naszym krótkim poradniku pedagogicznym](#). Regularne i konsekwentne stosowanie precyzyjnego, technicznego słownictwa przez nauczycieli może znacząco wspierać rozumienie pojęciowe uczniów.

Do opracowania wyjaśnień wykorzystaliśmy teorię „fal semantycznych”. Każda definicja składa się z trzech części: pierwsza część to wyjaśnienie terminu w bardziej abstrakcyjny sposób, druga część rozwija znaczenie terminu za pomocą powszechnego przykładu, a trzecia zawiera podsumowanie tego, co zostało wyjaśnione na przykładzie, w bardziej abstrakcyjnych kategoriach, aby ponownie nawiązać do słownictwa. Dodatkowe informacje na temat fal semantycznych znajdziesz w [naszym krótkim poradniku pedagogicznym](#).

Jest to wersja 1 słownika. Słownik ten będzie uzupełniany, poprawiany i aktualizowany wraz z rozwojem lekcji Experience AI.

Spis treści

[Cecha ML](#)
[Cykl życia projektu AI](#)
[Czyszczenie danych](#)
[Dane testowe ML](#)
[Dane treningowe modelu ML](#)
[Dane](#)
[Dokładność ML](#)
[Drzewo decyzyjne ML](#)
[Etykieta ML](#)
[Generatywna AI](#)
[Karta modelu ML](#)
[Klasa ML](#)
[Klasyfikacja ML](#)
[Model ML](#)
[Próg ufności ML](#)
[Przewidywanie ML](#)
[Systemy oparte na danych](#)
[Systemy oparte na regułach](#)
[Sztuczna inteligencja](#)
[Tendycyjność danych](#)
[Tendycyjność](#)
[Trenowanie modelu ML](#)
[Uczenie maszynowe](#)
[Uczenie nadzorowane](#)
[Uczenie nienadzorowane](#)
[Uczenie przez wzmacnianie](#)
[Ufność ML](#)
[Umiejętność korzystania z AI](#)
[Uprzedzenia społeczne](#)
[Węzeł drzewa decyzyjnego ML](#)
[Widzenie komputerowe](#)
[Wyjaśnialność ML](#)

Cecha ML

W **uczeniu maszynowym (ML)** cechy reprezentują właściwości skojarzone z **danymi**. Na przykład cechami zestawu danych muzycznych mogą być tempo, wysokość dźwięku, energia czy gatunek. Niektóre **modele** ML są **trenowane** przy użyciu cech w celu wykrycia podobieństw danych. Z kolei inne modele mogą **przewidywać** nowe cechy danych, które są niedostrzegalne dla człowieka. Wybór cech używanych podczas trenowania modelu ML może mieć znaczący wpływ na jego skuteczność.

Cykl życia projektu AI

Cykl życia projektu **sztucznej inteligencji (AI)** odnosi się do różnych etapów, które mogą być potrzebne do zaprojektowania i utworzenia **modelu uczenia maszynowego (ML)**. Etapy te obejmują zdefiniowanie problemu, przygotowanie **danych**, **trenowanie** modelu, testowanie i ocenę modelu oraz jego wyjaśnienie. Załóżmy, że projektujemy model ML służący do generowania nowej playlisty z utworami muzycznymi. Programista ML decyduje najpierw o tym, jaki typ playlisty utworzy. Następnie zbiera i przygotowuje dane dotyczące utworów muzycznych. Na podstawie tych danych trenuje i testuje model ML. Następnie model ML poddawany jest ocenie w celu sprawdzenia, czy działa zgodnie z oczekiwaniami. Na koniec wyjaśnia się jego działanie, aby inne osoby mogły z niego korzystać. Zwykle etapy cyklu życia projektu AI stosowane są w sposób iteracyjny, a nie jeden po drugim. Cykl życia projektu AI to seria iteracyjnych kroków wykorzystywanych do tworzenia i ulepszania modelu ML.

Czyszczenie danych

Czyszczenie danych to etap przygotowywania **danych** używanych do **trenowania modelu uczenia maszynowego (ML)**. Polega na wykrywaniu i korygowaniu błędów w danych. Na przykład poprawianie błędów w pisowni lub usuwanie duplikatów w danych tekstowych to dwa najprostsze zadania oczyszczania danych. Dane są bardzo często chaotyczne i wymagają bardziej złożonego procesu oczyszczania przed zastosowaniem ich do trenowania modeli ML. Istnieje wiele sposobów czyszczenia danych, w zależności od problemu i typu danych. Użycie czystych danych jest kluczowe dla tworzenia **dokładnych** modeli ML.

Dane testowe ML

W **uczeniu maszynowym (ML)** dane testowe odnoszą się do **danych** używanych do testowania i oceny **wytrenowanych modeli** ML. Załóżmy, że model ML został wytrenowany do **przewidywania** diagnozy stanu medycznego. Przed zastosowaniem w rzeczywistych sytuacjach testuje się model i ocenia go przy użyciu danych testowych. Dane testowe są oddzielone od **danych treningowych**, na podstawie których trenowany jest model. Służą do pomiaru wydajności modelu ML w oparciu o przykłady, które nie były uwzględnione w danych treningowych.

Dane treningowe modelu ML

W **uczeniu maszynowym (ML)** dane treningowe to przykłady w postaci **danych**, które wykorzystuje się do **trenowania modeli** ML. Programiści ML tworzą modele, aby opracować wzorce w danych treningowych, które można wykorzystać do generowania **przewidywań** dotyczących nowych danych. Na przykład programista ML tworzy aplikację służącą do rozpoznawania mowy. Dane treningowe mogą zawierać wiele przykładów wypowiedzi osób mówiących z różnym akcentem i posługujących się różnym tonem głosu. Im dokładniej dane treningowe odzwierciedlają rzeczywistość, tym lepsze wyniki osiągnie model.

Dane

Termin „dane” odnosi się do wartości, pomiarów, faktów lub obserwacji przedstawionych w formie odpowiedniej do przetwarzania przez programy komputerowe. Istnieje wiele typów danych, takich jak tekst, obraz lub dźwięk. Przykładem danych tekstowych są wiadomości wymieniane między znajomymi na urządzeniach cyfrowych. W **uczeniu maszynowym (ML)** dane reprezentują przykłady, na podstawie których **modele ML** są **trenowane**. Zbieranie, **czyszczenie** i strukturyzowanie ogromnych ilości danych to istotny element projektowania modeli uczenia maszynowego.

Dokładność ML

Dokładność oznacza, w jakim stopniu coś jest poprawne. W **uczeniu maszynowym (ML)** dokładność mierzy, jak często **model ML** dokonuje poprawnych **przewidywań**. Załóżmy, że zadaniem modelu **klasyfikacji** jest klasyfikowanie jabłek. Spośród 100 obrazów jabłek, 90 zostało sklasyfikowanych poprawnie. Dokładność klasyfikacji modelu wynosi zatem 90%. Dokładność jest jednym z kryteriów oceny modeli ML. Najczęściej stosuje się ją w połączeniu z innymi miarami, aby ocenić jakość modelu.

Drzewo decyzyjne ML

Drzewo decyzyjne w **uczeniu maszynowym (ML)** to jeden z typów **modelu ML**. Programiści ML używają drzew decyzyjnych do opracowania struktury zestawu warunków, na podstawie których można dokonać **przewidywania**. Warunki te są wyznaczane na podstawie **cech** występujących w **danych**. Na przykład drzewo decyzyjne można wykorzystać do utworzenia systemu rekomendacji filmów. Model drzewa decyzyjnego jest **trenowany** przy użyciu preferencji filmowych wielu osób. W trakcie treningu generowane są warunki na podstawie takich cech, jak typ filmu, długość filmu czy główny aktor. Następnie model ML przewiduje, jaki film dana osoba chciałaby obejrzeć, analizując stopień dopasowania jej preferencji do warunków zastosowanych w modelu. Struktura drzew decyzyjnych ML jest generowana na podstawie dużych ilości danych i może ulec zmianie, gdy model zostanie ponownie wytrenowany na podstawie innego zestawu danych.

Etykieta ML

W **uczeniu nadzorowanym model uczenia maszynowego (ML)** jest **trenowany** przy użyciu **danych** oznaczonych etykietami. Każdy fragment danych jest oznaczony jedną lub wieloma etykietami, które dostarczają informacji o tych danych. Na przykład model ML ma rozpoznawać dźwięki ptaków. Każdy dźwięk jest oznaczony nazwą ptaka, który wydaje dany dźwięk. Model ML jest trenowany na podstawie dźwięków oznaczonych etykietami i może **przewidywać** etykietę (nazwę ptaka) dla nowych dźwięków. Dane są najczęściej oznaczane przez człowieka, tak aby przykłady służące do trenowania modeli ML były jak najbardziej precyzyjne.

Generatywna AI

Generatywna AI to typ **sztucznej inteligencji (AI)** zaprojektowany do generowania treści, takich jak tekst, obrazy czy dźwięk. Istnieje wiele aplikacji wykorzystujących generatywną AI, między innymi aplikacje przeznaczone do tworzenia dzieł sztuki, utworów muzycznych oraz generowania tekstów dla chatbotów. Na przykład w aplikacjach do tworzenia dzieł sztuki przy użyciu generatywnej AI można stworzyć obraz na podstawie polecenia, np. „stwórz obraz smoka czytającego książkę”. Sztuka wykorzystująca generatywną AI jest tworzona przy użyciu **modeli uczenia maszynowego wytrenowanych** na podstawie milionów obrazów istniejących dzieł sztuki. Wygenerowane obrazy mogą odtwarzać styl artysty bez jego wiedzy czy zgody. Aplikacje generatywnej AI stają się coraz bardziej powszechne i często trudno jest zorientować się, kiedy są używane.

Karta modelu ML

Karta modelu **uczenia maszynowego (ML)** to sposób dokumentowania istotnych informacji o **modelach** ML w uporządkowany sposób. Karty modeli ML są tworzone przez programistów ML i przeznaczone zarówno dla ekspertów, jak i dla osób niebędących ekspertami. Załóżmy, że zadaniem aplikacji ML jest tłumaczenie tekstów na różne języki, np. z arabskiego na francuski i odwrotnie. Karta modelu zawiera informacje o **dokładności** tłumaczenia wykonanego przez model, a także o wydajności modelu w zakresie rozpoznawania żargonu, slangu i dialektów. Inne informacje zawarte w karcie modelu mogą dotyczyć typu modelu ML, różnych wskaźników wydajności, a nawet znanej **tendencji**. W **cyklu życia projektu AI** karty modelu tworzy się na etapie wyjaśniania, aby w przejrzysty sposób przedstawić informacje dotyczące możliwości oraz ograniczeń modelu.

Klasa ML

Aby **trenować modele klasyfikacyjne**, twórcy modeli **uczenia maszynowego (ML)** grupują **dane** w predefiniowane grupy zwane „klasami”. Są one z góry zdefiniowane na podstawie kryteriów, które dana grupa osób uzna za najbardziej przydatne. Załóżmy, że aplikacja ML ma rozpoznawać owoce w supermarkecie. Dane mogą być pogrupowane w klasy obejmujące jabłka, banany, pomarańcze, jagody itp. Klasa to grupa elementów, którą modele klasyfikacyjne wykorzystują do rozpoznawania wzorców w danych.

Klasyfikacja ML

Klasyfikacja to zadanie polegające na przypisywaniu elementów do wcześniej określonych grup, zwanych **klasami**. Klasy są z góry zdefiniowane na podstawie kryteriów, które dana grupa osób uzna za najbardziej przydatne. Przykładem problemu klasyfikacji jest analiza sentymentu w recenzjach piosenek. **Model** klasyfikacyjny **uczenia maszynowego (ML)** jest **trenowany** przy użyciu recenzji **oznaczonych** przez ich autorów etykietą „negatywne” lub „pozytywne”. Wytrenowanego w ten sposób modelu można użyć do **przewidywania**, czy nowa recenzja powinna zostać sklasyfikowana jako „negatywna” czy „pozytywna”. Model klasyfikacyjny przewidyuje jedną lub wiele

etykiet klas. Podejście klasyfikacyjne jest przydatne do rozwiązywania problemów, w przypadku których odpowiedź mieści się w predefiniowanych grupach.

Model ML

Model **uczenia maszynowego (ML)** jest używany w aplikacji ML do wykonywania zadań lub rozwiązywania problemów. Model ML jest reprezentacją rozwiązywanego problemu. Programiści ML wykorzystują ogromne ilości **danych** reprezentatywnych dla konkretnego problemu, aby **wytrenować** model do wykrywania wzorców. W wyniku treningu powstaje model, który następnie jest używany do tworzenia **przewidywań** dotyczących nowych danych w tym samym kontekście. Na przykład samochody autonomiczne konstruuje się przy użyciu modeli ML, które przewidują, kiedy należy się zatrzymać. Modele takie są trenowane na milionach przykładów sytuacji, w których samochody powinny się zatrzymać. Istnieje wiele różnych typów modeli, wykorzystujących różnego rodzaju dane i różne metody treningu. Wszystkie z nich są trenowane w tym samym celu: wykrywania wzorców w **danych treningowych**, a następnie generowania przewidywań dotyczących nowych danych.

Próg ufności ML

Próg ufności to wartość ustawiona jako poziom akceptacji **przewidywań modelu uczenia maszynowego (ML)**. Próg ufności jest określany przez twórcę ML podczas projektowania modelu ML. Na przykład model ML może przewidywać z 50% **ufnością**, że jutro będzie śnieżyca. Jeśli jednak ustalono próg ufności na poziomie 60%, takie przewidywanie zostanie uznane za niedokładne. Mówiąc inaczej, aby przewidywanie zostało uznane za **dokładne**, jego ufność musi wynosić 60% lub więcej. Wartość jest ustalana w zależności od charakteru rozwiązywanego problemu – przewidywanie diagnozy medycznej wymaga wyższego progu ufności niż przewidywanie rekomendacji piosenek. Wybór wartości progowej decyduje o tym, jaki jest akceptowalny poziom ufności przewidywania.

Przewidywanie ML

Modele uczenia maszynowego (ML) są **trenowane** w celu generowania przewidywań. Przewidywanie wygenerowane przez model ML sugeruje, co **dane** reprezentują lub co może być przydatne w zadaniu. Na przykład programista ML może wytrenować model tak, aby przewidywał, które filmy mogą zainteresować daną osobę na podstawie jej dotychczasowych preferencji filmowych. Model wygeneruje prognozę po wytrenowaniu go na podstawie danych dotyczących wyborów filmowych wielu osób. Głównym zadaniem modelu ML jest generowanie przewidywań. Wszystkie modele ML generują przewidywania, nawet jeśli w niektórych przypadkach są one trudne do zauważenia przez użytkownika.

Systemy oparte na danych

W projektowaniu systemów opartych na danych wykorzystuje się **dane** zamiast instrukcji krok po kroku. Na przykład ustalenie przyczyn niektórych chorób jest trudne, istnieje jednak wiele przykładowych danych. Projektanci wykorzystują dane medyczne osób dotkniętych daną chorobą

do diagnozowania jej. Systemy oparte na danych różnią się od systemów **opartych na regułach**. Są odpowiednie do rozwiązywania problemów, w których trudno jest opracować reguły uwzględniające każdą sytuację. Zamiast tego można zebrać wystarczającą liczbę przykładów pozwalającą znaleźć rozwiązanie.

Systemy oparte na regułach

Systemy oparte na regułach projektuje się przy użyciu zestawu predefiniowanych reguł. Na przykład program do gry w kółko i krzyżyk można zaprojektować za pomocą reguł określających, jakie ruchy należy wykonać, aby wygrać grę. Reguły definiowane są przez osoby będące ekspertami w dziedzinie, której dotyczy rozwiązywany problem. Systemy **sztucznej inteligencji (AI)** oparte na regułach często określa się również mianem „tradycyjnej AI”. Są one przeciwieństwem **systemów opartych na danych**, w których wykorzystuje się **dane** jako przykładowe rozwiązania danego problemu. Systemy oparte na regułach są przydatne do rozwiązywania problemów, w których można opracować i zastosować reguły obejmujące większość sytuacji.

Sztuczna inteligencja

Sztuczna inteligencja (AI) to projektowanie i badanie systemów, które wykazują cechy inteligentnego zachowania. Niektóre aplikacje AI opierają się na regułach. Obecnie coraz częściej aplikacje AI są tworzone przy użyciu **uczenia maszynowego**. Uważa się, że ma ono zdolność „uczenia się” z przykładów, które zawarte są w **danych**. Na przykład niektóre aplikacje AI zostały opracowane w celu udzielania odpowiedzi na pytania lub diagnozowania chorób. Z kolei inne zostały utworzone w szkodliwych celach, takich jak rozpowszechnianie fałszywych informacji. Aplikacje AI nie myślą samodzielnie. Są projektowane tak, aby wykonywać zadania w sposób, który wydaje się inteligentny.

Tendencyjność danych

Tendencyjność danych odnosi się do **tendencyjności** widocznej w **danych** używanych do **trenowania modeli uczenia maszynowego (ML)**. Może to prowadzić do generowania tendencyjnych **przewidywań** przez modele ML. Na przykład niektóre modele rozpoznawania twarzy są tendencyjne wobec twarzy o określonym kolorze skóry, ponieważ zostały wytrenowane głównie na podstawie obrazów twarzy o jednym kolorze skóry. Istnieje kilka potencjalnych źródeł tendencyjności danych. Należą do nich niekompletne dane lub dane odzwierciedlające **uprzedzenia społeczne**. Wykrywanie tendencyjności danych jest istotne, ponieważ pozwala uniknąć generowania przez modele ML tendencyjnych przewidywań.

Tendencyjność

Tendencyjność oznacza skłonność do preferowania lub odrzucania czegoś. Na przykład uczeń woli lekcje języka angielskiego od lekcji matematyki i poświęca więcej czasu na odrabianie zadań domowych z języka angielskiego. Ma zatem skłonność, czy też tendencję, do nauki języka angielskiego. Istnieje wiele rodzajów tendencyjności, na przykład **tendencyjność danych** czy **uprzedzenia społeczne**. Twórcy modeli **uczenia maszynowego (ML)** muszą dokładnie

przeanalizować, czy **dane** wykorzystywane do **trenowania modeli ML** nie są tendencyjne. Tendencyjność może prowadzić do faworyzowania jednej osoby, grupy lub zestawu idei i przekonań kosztem innych.

Trenowanie modelu ML

Modele uczenia maszynowego (ML) trenuje się na podstawie przykładów w postaci **danych**, aby „nauczyć je” wyszukiwania wzorców i generowania **przewidywań**. W trakcie treningu modele są dostosowywane, aby poprawić jakość generowanych przewidywań. Na przykład programista ML tworzy model służący do rekomendowania piosenek. Model ML będzie trenowany na podstawie wyborów muzycznych wielu osób, aby znaleźć podobieństwa między ich preferencjami muzycznymi. Im bardziej zróżnicowane utwory muzyczne zostaną wybrane do trenowania modelu, tym lepsze będą przewidywane rekomendacje utworów. Istnieje wiele różnych sposobów trenowania modeli ML, z wykorzystaniem różnego typu danych. Programista wybiera odpowiedni typ treningu w zależności od rozwiązywanego problemu oraz dostępnych danych. Jakość treningu zależy w dużej mierze od jakości wykorzystanych danych.

Uczenie maszynowe

Uczenie maszynowe (ML) to metoda wykorzystywana do projektowania i tworzenia systemów **sztucznej inteligencji (AI)**. Zamiast wykonywać instrukcje krok po kroku, ML „uczy się”, wykorzystując przykłady w postaci **danych**. Mówiąc inaczej, aplikacje ML są **oparte na danych**. Na przykład aplikacja ML ma rozpoznawać mowę. Oparta jest na wielu przykładach wypowiedzi osób mówiących z różnym akcentem i posługujących się różnym tonem głosu. Zadaniem innych aplikacji ML może być rozpoznawanie obiektów na obrazach, czy granie w skomplikowane gry. Każda aplikacja ML projektowana jest do rozwiązywania konkretnego problemu.

Uczenie nadzorowane

Uczenie nadzorowane to jedna z metod **trenowania modeli uczenia maszynowego (ML)**. W uczeniu nadzorowanym wykorzystuje się duże ilości **danych oznaczonych etykietami** przez osoby posiadające odpowiednie informacje. Jednym z typów uczenia nadzorowanego jest **klasyfikacja**. Przykładem problemu klasyfikacji może być identyfikacja tygrysów w środowisku naturalnym. Dane składają się z wielu obrazów, a obrazy zawierające tygrysy są oznaczone odpowiednimi etykietami. Model ML jest trenowany na podstawie oznaczonych obrazów i **przewiduje**, czy znajduje się na nich tygrys. Poprawnie oznaczone obrazy pozwalają programiście ocenić, w jakim stopniu przewidywania modelu są **dokładne** i dostosować jego trenowanie. Następnie można zastosować model ML do przewidywania, czy na zupełnie nowych obrazach znajduje się tygrys. Metoda uczenia nadzorowanego opiera się na wystarczającej ilości poprawnie oznaczonych danych umożliwiających generowanie dokładnych przewidywań.

Uczenie nienadzorowane

Uczenie nienadzorowane to jedna z metod **trenowania modeli uczenia maszynowego (ML)**. W tej metodzie programiści ML trenują modele w taki sposób, aby **dane** były grupowane na podstawie

podobieństw. Proces ten pozwala na wykrycie ukrytych wzorców w danych. Jednym z typów uczenia nadzorowanego jest klastrowanie (grupowanie). Przykładem problemu klastrowania jest **przewidywanie**, w jaki sposób można pogrupować dane dotyczące stanu zdrowia w celu diagnozowania chorób. Takie grupy danych nazywane są klastrami i nie są one z góry określone. Modelu ML można użyć do przewidywania, czy nowe dane dotyczące stanu zdrowia należą do jednego z klastrów. Metoda uczenia nienadzorowanego może być użyteczna w rozwiązywaniu problemów, w przypadku których nie jest jasne, czego się szuka.

Uczenie przez wzmacnianie

Uczenie przez wzmacnianie to jedna z metod **trenowania modeli uczenia maszynowego (ML)**. Metoda ta używana jest w przypadku rozwiązywania problemów z jasno określonym celem, do którego osiągnięcia stosuje się system nagród i kar. Strategie oparte na uczeniu przez wzmacnianie są wykorzystywane do projektowania samochodów autonomicznych lub skomplikowanych gier. Model tego typu można zastosować na przykład do zaprojektowania aplikacji do gry w szachy. Model trenuje się w taki sposób, aby **przewidywał** ruchy, które przynoszą największe korzyści i minimalizują straty, prowadząc do wygranej. W metodzie uczenia przez wzmacnianie wykorzystuje się nagrody i kary do wyznaczania strategii umożliwiającej osiągnięcie określonego celu.

Ufność ML

Ufność odnosi się do stopnia pewności. W **uczeniu maszynowym (ML)** ufność to sposób pomiaru pewności **przewidywania**. Na przykład **model klasyfikacyjny** ma przewidywać, czy jutro będzie padać deszcz. Model przewiduje z ufnością na poziomie 90%, że jutro będzie padać. Mówiąc inaczej, jest na 90% pewne, że jutro będzie deszcz. Zastosowanie ufności do pomiaru stopnia pewności przewidywań ułatwia ocenę jakości modelu ML.

Umiejętność korzystania z AI

Umiejętność korzystania z AI to zbiór kompetencji i umiejętności poznawczych, które pozwalają ludziom w świadomy sposób korzystać z aplikacji **sztucznej inteligencji (AI)**, a także rozpoznawać sytuacje, w których aplikacje AI są używane w ich otoczeniu. Kompetencje te obejmują rozumienie działania AI, uczestnictwo w rozwoju systemów AI oraz posiadanie świadomych opinii na temat sposobów, w jakie systemy AI są wykorzystywane na świecie. Na przykład dana osoba ma umiejętność korzystania z AI, gdy potrafi ocenić dokładność informacji dostarczanych przez aplikację czatbota opartą na AI. Oprócz nawiązywania różnego typu interakcji z aplikacjami AI umiejętność korzystania z AI obejmuje także zdolność danej osoby do aktywnego uczestnictwa w decydowaniu o tym, w jaki sposób systemy AI mogą być używane w jej otoczeniu.

Uprzedzenia społeczne

Uprzedzenia społeczne to **uprzedzenia**, które podziela znaczna grupa osób lub całe społeczeństwo. Istnieje wiele różnych rodzajów uprzedzeń społecznych, takich jak uprzedzenia rasowe, etniczne, czy uprzedzenia związane z płcią. Przykładem uprzedzeń związanych z płcią jest przekonanie, że kobiety są mniej predysponowane do wykonywania zawodów technicznych niż mężczyźni. **Dane**

zebrane od dużej grupy osób mogą odzwierciedlać uprzedzenia społeczne, prowadząc do **tendencji danych**. Wykorzystanie danych odzwierciedlających uprzedzenia społeczne do **trenowania modeli uczenia maszynowego (ML)** może prowadzić do generowania przez modele tendencyjnych **przewidywań**. Ograniczanie wszelkich przejawów uprzedzeń społecznych w **danych treningowych** jest bardzo istotne, ponieważ pozwala uniknąć generowania dyskryminujących lub niesprawiedliwych wyników.

Węzeł drzewa decyzyjnego ML

Drzewo decyzyjne w **uczeniu maszynowym (ML)** składa się z węzłów. Są one ze sobą połączone, tworząc strukturę, na podstawie której można wygenerować **przewidywanie**. Istnieją dwa typy węzłów: węzły decyzyjne i węzły liścia. Rozważmy na przykład drzewo decyzyjne utworzone w celu przewidywania typów gwiazd w naszym Układzie Słonecznym. Węzły decyzyjne reprezentują **cechy danych**, takie jak temperatura, promień, kolor lub jasność gwiazd. Węzły liścia reprezentują typy gwiazd w formie **etykiet** przewidywań, np. „Czerwony Karzeł”, „Biały Karzeł” czy „Brązowy Karzeł”. Węzły drzewa decyzyjnego tworzą strukturę niezbędną do generowania przewidywań przez **model ML**.

Widzenie komputerowe

Widzenie komputerowe to dziedzina zajmująca się systemami przeznaczonymi do przetwarzania informacji z cyfrowych obrazów lub filmów. Przykłady zastosowań widzenia komputerowego obejmują rozpoznawanie twarzy, obrazowanie medyczne oraz monitoring wizyjny. Na przykład widzenie komputerowe jest używane w projektowaniu samochodów autonomicznych do wykrywania i unikania kolizji z obiektami. Najczęściej systemy widzenia komputerowego wykorzystują **modele uczenia maszynowego** do identyfikowania wzorców w danych pochodzących z obrazów i filmów. Systemy widzenia komputerowego są przydatne w przypadku, gdy informacje z cyfrowych obrazów lub filmów można wykorzystać do rozwiązania problemu.

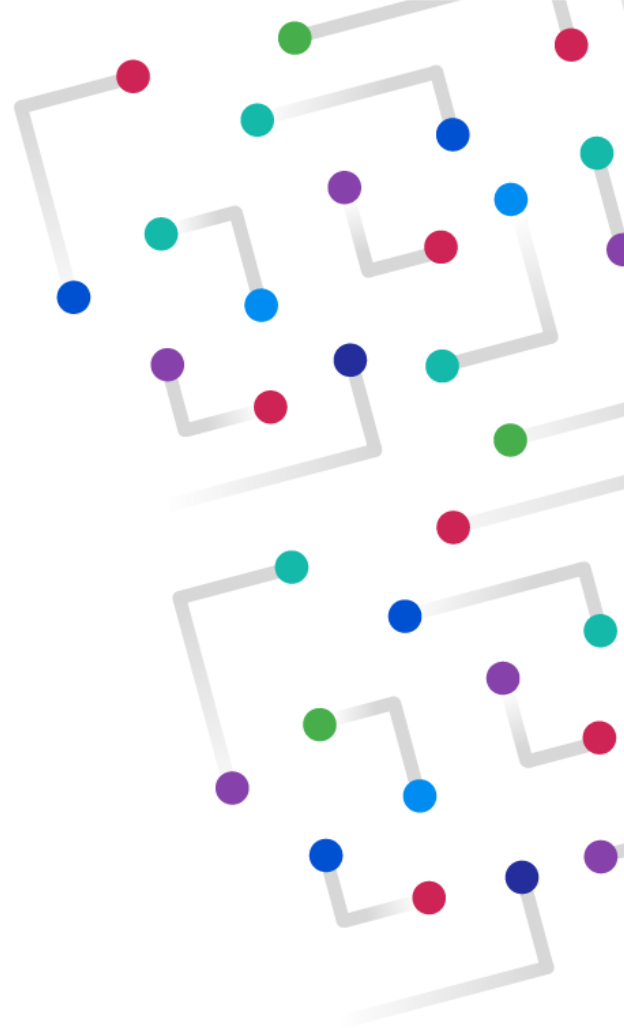
Wyjaśnialność ML

Wyjaśnialność oznacza, w jakim stopniu coś jest zrozumiałe. W **uczeniu maszynowym (ML)** wyjaśnialność pozwala człowiekowi zrozumieć, w jaki sposób model wygenerował **przewidywanie**. Na przykład **modele ML drzewa decyzyjnego** są wyjaśnialne, ponieważ można przeanalizować ich **węzły** w sposób zrozumiały dla człowieka. Działania większości modeli ML nie można w pełni wyjaśnić, a niektóre z nich są łatwiejsze do wyjaśnienia niż inne. Zwiększenie wyjaśnialności modelu może ułatwić rozwiązywanie problemów i zwalczanie **tendencji**.



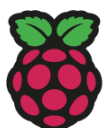
Ten zasób jest objęty licencją [Raspberry Pi Foundation](#) na podstawie Międzynarodowej Licencji Publicznej Creative Commons Uznanie autorstwa-Użycie niekomercyjne-Bez utworów zależnych 4.0 (CC BY-NC-ND 4.0). For more information on this licence, see creativecommons.org/licenses/by-nc-nd/4.0.

Experience AI



Słownik został opracowany przez Raspberry Pi Foundation w ramach programu edukacyjnego Experience AI. Więcej informacji na stronie www.experience-ai.org

Ten zasób jest objęty licencją Raspberry Pi Foundation na podstawie Międzynarodowej Licencji Publicznej Creative Commons Uznanie autorstwa-Użycie niekomercyjne-Bez utworów zależnych 4.0 (CC BY-NC-ND 4.0). Więcej informacji na temat tej licencji można znaleźć na stronie creativecommons.org/licenses/by-nc-nd/4.0.



Raspberry Pi
Foundation