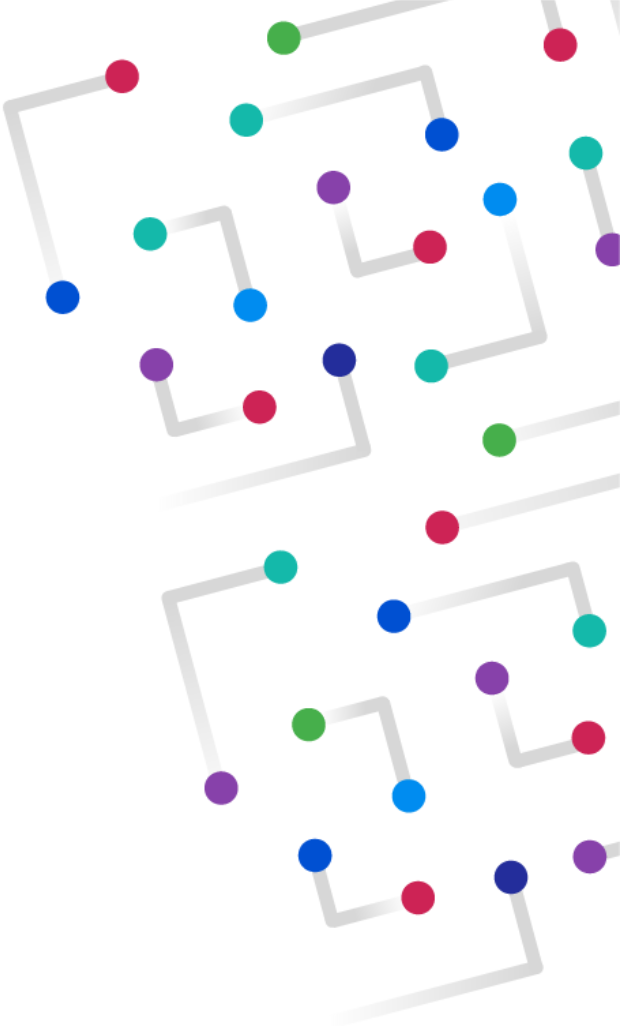
Raspberry Pi Foundation

Experience AI

Glosar de termeni despre AI



Experience AI — Glosar de termeni

Acest glosar explică o serie de termeni-cheie din domeniul inteligenței artificiale (AI) și al învățării automate (ML) folosiți în lecțiile Experience AI și nu numai.

Mergi la glosar

Aceste explicații au fost concepute în primul rând pentru profesori și educatori și se adresează unui public tânăr. Acest glosar își propune să te ajute să înțelegi mai bine acești termeni-cheie și să-ți îmbunătățești cunoștințele tehnice.

Vocabularul constituie o parte importantă a predării și învățării. Folosirea corectă a vocabularului îi poate ajuta pe cei care învață să înțeleagă materia mai bine. Pe de altă parte, utilizarea inconsecventă a termenilor poate duce la fixarea unor concepții alternative (numite adesea concepții eronate), care pot afecta învățarea. Poți citi mai multe despre asta în Ghidul rapid despre concepțiile alternative. Ca profesor, e important să utilizezi termenii tehnici corect, în mod regulat și consecvent pentru ca elevii să înțeleagă materia la nivel conceptual.

Pentru a elabora aceste explicații, am folosit teoria „undelor semantice”. Fiecare explicație urmează aceeași structură în trei părți: prima parte este o definiție mai abstractă a termenului; a doua explică sensul acestuia folosind un exemplu comun, iar a treia parte reia explicația într-o manieră mai abstractă, pentru a face din nou conexiunea cu termenul definit. Poți afla mai multe din Ghidul rapid despre undele semantice.

Aceasta este versiunea 1 a glosarului. Acest glosar va fi completat, revizuit și actualizat odată cu evoluția lecțiilor Experience AI.

Cuprins

Inteligență artificială (AI)
Învățare automată (ML)
Alfabetizare AI
AI generativă
Antrenament
Caracteristică
Ciclul de viață al unui proiect AI
Clase ML
Clasificarea ML
Computer vision
Curățarea datelor
Date
Data-driven (bazat pe date)
Date de testare
Date de antrenament
Etichetă
Învățare prin consolidare
Învățare supravegheată
Învățare nesupravegheată
Acuratețe (ML)
Caracter explicabil
Grad de încredere
Prag de încredere
Model ML
Fișă de model
Predicție
Prejudecată (non-imparțialitate)
Prejudecăți în date
Prejudecată socială
Rule-based (bazat pe reguli)

Inteligență artificială (AI)

Inteligența artificială (AI) reprezintă proiectarea și studiul sistemelor care par să imite un comportament inteligent. Unele aplicații AI se bazează pe reguli. În prezent, aplicațiile AI sunt mai frecvent construite folosind **învățarea automată**, despre care se poate spune că „învață” din exemple sub formă de **date**. De exemplu, unele aplicații AI sunt create pentru a răspunde la întrebări sau pentru a ajuta la diagnosticarea bolilor. Alte aplicații AI ar putea fi create în scopuri dăunătoare, cum ar fi răspândirea de știri false. Aplicațiile AI nu gândesc. Ele sunt create pentru a îndeplini sarcini într-un mod care pare a fi produsul inteligenței.

Învățare automată (ML)

Învățarea automată (ML) este o abordare utilizată pentru a proiecta și construi sisteme de **inteligență artificială (AI)**. Se spune că ML „învață” folosind exemple sub formă de **date**, în loc să execute instrucțiuni pas cu pas. Cu alte cuvinte, aplicațiile ML sunt **data-driven (bazate pe date)**. Să luăm de exemplu o aplicație ML care este utilizată pentru recunoașterea vorbirii. Aceasta se bazează pe multe exemple de oameni care vorbesc cu diferite accente și pe diferite tonuri. Alte aplicații ML includ identificarea obiectelor din imagini sau jocurile complexe. Fiecare aplicație ML este concepută pentru a rezolva o problemă anume.

Alfabetizare AI

Alfabetizarea AI reprezintă un set de competențe și moduri de gândire care le permit oamenilor să interacționeze în moduri care au sens cu aplicațiile de **inteligență artificială(AI)**, precum și în alte situații în care se întâlnesc cu aplicații AI. Astfel de competențe includ înțelegerea inteligenței artificiale, implicarea în dezvoltarea sistemelor AI și faptul de a avea opinii avizate cu privire la modurile în care sunt utilizate în lume sistemele AI. De exemplu, alfabetizarea AI este o competență utilă atunci când evaluezi acuratețea informațiilor furnizate de o aplicație AI de chatbot. Pe lângă diferitele modalități de a interacționa cu aplicațiile AI, alfabetizarea AI include și capacitatea oamenilor de a participa în mod activ la decizii privind utilizarea sistemelor AI în general.

AI generativă

AI generativă este un tip de **inteligență artificială (AI)** concepută pentru a genera conținut de tip text, vizual sau audio. Există o mulțime de aplicații care folosesc AI generativă, inclusiv producția de artă sau muzică, ori generarea de text pentru chatboți. De exemplu, aplicațiile artistice ce folosesc AI generativă pot genera o imagine pe baza unei comenzi, cum ar fi de pildă „generează-mi o imagine cu un dragon citind o carte”. Lucrările artistice realizate cu AI generativă sunt create folosind **modele** de **învățare automată antrenate** pe milioane de imagini existente cu opere de artă. Imaginile rezultate pot reproduce stilul unui artist fără știrea sau aprobarea acestuia. Aplicațiile AI generativă sunt din ce în ce mai des întâlnite și adesea este imposibil să îți dai seama că a fost folosită AI generativă.

Antrenament

Modelele de **învățare automată (ML)** sunt antrenate folosind exemple sub formă de **date** pentru a identifica tipare regulate și a face **predicții**. În timpul antrenamentului, modelele sunt ajustate pentru a îmbunătăți predicțiile. De exemplu, un dezvoltator ML ar putea construi un model pentru a le recomanda melodii utilizatorilor. Modelul ML va fi instruit folosind preferințele muzicale ale unui număr mare de oameni, pentru a descoperi asemănări între preferințele unor oameni diferiți. Cu cât modelul este antrenat cu preferințe cât mai diverse, cu atât mai bune vor fi recomandările. Există multe moduri diferite de antrenare a modelelor ML, folosind diferite tipuri de date. Un dezvoltator va alege între tipurile disponibile de antrenament în funcție de problema pe care încearcă să o rezolve și de datele disponibile pentru a o rezolva. Calitatea antrenării modelului depinde în mare măsură de calitatea datelor utilizate.

Caracteristică

În **învățarea automată (ML)**, caracteristicile sunt asociate **datelor**. De exemplu, un set de date muzicale poate avea caracteristici precum tempo, înălțime, energie sau gen muzical. Unele **modele** ML sunt **antrenate** folosind diverse caracteristici pentru a descoperi similarități în date. Alte modele **prezic** existența a noi caracteristici ale datelor, caracteristici pe care oamenii nu le pot observa cu ușurință. Alegerea caracteristicilor de utilizat pentru antrenarea unui model ML poate fi decisivă pentru funcționarea corectă a modelului.

Ciclul de viață al unui proiect AI

Ciclul de viață al unui proiect de **inteligență artificială (AI)** constă în diferitele etape ale proiectării și dezvoltării unui **model** de **învățare automată (ML)**. Aceste etape includ definirea problemei, pregătirea **datelor**, **antrenarea** modelului, testarea, evaluarea și explicarea acestuia. Să luăm ca exemplu un model ML conceput pentru a genera noi playlist-uri de melodii. Un dezvoltator ML ar putea lua mai întâi în considerare ce tip de playlist dorește să creeze. Apoi, ar putea colecta și pregăti datele legate de melodii. Modelul ML este antrenat și testat pe baza acestor date. Apoi, modelul este evaluat pentru a vedea dacă funcționează conform așteptărilor. În cele din urmă, modelul ML este explicat pentru a putea fi folosit și de alții. De obicei, etapele ciclului de viață al unui proiect AI sunt parcurse mai degrabă repetitiv decât una după alta. Ciclul de viață al unui proiect AI este o serie de etape repetitive care sunt parcurse pentru a construi și îmbunătăți un model ML.

Clase ML

Pentru a **antrena modelele de clasificare**, dezvoltatorii de software de **învățare automată (ML)** organizează **datele** în grupuri predefinite denumite clase. Clasele sunt definite în prealabil după anumite criterii. Să ne imaginăm o aplicație ML concepută pentru a identifica fructele într-un supermarket. Datele pot fi organizate în clase de: mere, banane, portocale, afine etc. O clasă este un grup de lucruri pe care modelele de clasificare îl folosesc pentru a identifica asemănări în date care sunt nestructurate.

Clasificarea ML

Clasificarea se referă la acțiunea de a grupa lucruri în grupuri predefinite, numite **clase**. Clasele sunt definite în prealabil în funcție de criterii considerate adecvate în acest scop. Un exemplu de problemă de clasificare este analiza tonului în recenziile pieselor muzicale. Un **model** de clasificare **ML** este **antrenat** pe baza recenziilor **etichetate** de oameni ca fiind „pozitive” sau „negative”. După antrenament, modelul ML poate fi utilizat pentru a **prezice** dacă o nouă recenzie ar trebui să fie clasificată ca „pozitivă” sau „negativă”. Un model de clasificare prezice una sau mai multe caracteristici (etichete) de clasă. O abordare de tipul clasificării este utilă pentru rezolvarea problemelor în care răspunsul ține de niște grupuri predefinite.

Computer vision

Computer vision (numită uneori și viziune computerizată) constă în studiul sistemelor concepute pentru a procesa informații conținute în imagini sau videoclipuri digitale. Printre exemplele de aplicații de computer vision se numără recunoașterea facială, imagistica medicală și supravegherea video. De exemplu, computer vision este utilizată în proiectarea mașinilor cu conducere autonomă, pentru a

detecta obstacole și a evita coliziunile. Cel mai adesea, sistemele de computer vision utilizează **modele** de **învățare automată** pentru a identifica tipare în date de tip imagine și video. Sistemele de computer vision sunt utile acolo unde informațiile digitale de tip foto sau video pot fi folosite pentru a rezolva o anumită problemă.

Curățarea datelor

Curățarea datelor constituie o etapă a pregătirii datelor folosite la **antrenarea** unui **model** de **învățare automată (ML)**. Curățarea datelor constă în identificarea și corectarea erorilor din date. De exemplu, remedierea erorilor de tastare sau eliminarea duplicatelor din datele de tip text sunt două exemple simple de curățare a datelor. Cel mai adesea, datele sunt în stare brută și necesită o curățare mai complexă înainte de a fi utilizate la antrenarea modelelor ML. Există multe modalități de curățare a datelor, în funcție de problemă și de tipul de date. Utilizarea datelor curate este esențială pentru construirea unor modele ML **precise**.

Date

Termenul de „date” se referă la valori, măsurători, fapte sau observații, prezentate într-o formă ce poate fi procesată de către computere. Există multe tipuri de date, cum ar fi date de tip text, imagine sau sunet. Un exemplu de date text sunt mesajele pe care oamenii le schimbă cu prietenii lor prin intermediul dispozitivelor digitale. În **învățarea automată (ML)**, datele reprezintă exemplele pe baza cărora sunt **antrenate modelele** ML. Colectarea, **curățarea** și structurarea unor cantități mari de date reprezintă o parte esențială a proiectării modelelor ML.

Data-driven (bazat pe date)

Data-driven (bazat pe date) este o expresie referitoare la o modalitate de proiectare a sistemelor folosind **date** în loc de instrucțiuni pas cu pas. De exemplu, este greu de determinat ce cauzează anumite boli, dar există o mulțime de exemple de cazuri. Prin urmare, dezvoltatorii folosesc datele medicale ale persoanelor afectate de boală pentru a o diagnostica. Sistemele bazate pe date contrastează cu cele **bazate pe reguli**. Sistemele bazate pe date sunt potrivite pentru rezolvarea acelor probleme în care este foarte greu să determini regulile care se aplică în fiecare situație particulară. În schimb, este posibil să se colecteze suficiente exemple pentru a construi o soluție.

Date de testare

În **învățarea automată (ML)**, datele de testare se referă la **datele** utilizate pentru testarea și evaluarea **modelelor** ML care sunt **antrenate**. De exemplu, un model ML este antrenat pentru a **prezice** un diagnostic medical. Înainte de a fi utilizat în situații reale, modelul este testat și evaluat folosind date de testare. Datele de testare sunt separate de **datele de antrenament** cu care este instruit un model ML. Datele de testare sunt folosite pentru a măsura performanța unui model ML folosind exemple diferite de datele de antrenament.

Date de antrenament

În **învățarea automată (ML)**, datele de antrenament se referă la exemplele sub forma de **date** utilizate la **antrenarea modelelor** ML. Dezvoltatorii ML construiesc modele pentru a identifica tipare în datele de antrenament, care pot fi apoi folosite pentru a genera **predicții** referitoare la date noi. Să luăm ca

exemplu un dezvoltator ML care creează o aplicație de recunoaștere a vorbirii. Datele de antrenament pot include foarte multe exemple de oameni care vorbesc, cu diferite accente sau pe diferite tonuri. Cu cât datele de antrenament reprezintă mai bine realitatea, cu atât mai probabil este ca modelul să funcționeze mai bine.

Etichetă

În **învățarea supravegheată**, un **model** de **învățare automată (ML)** este **antrenat** folosind **date** etichetate. Fiecărei informații individuale din volumul de date i se atribuie una sau mai multe etichete care oferă informații despre acele date. De exemplu, un model ML este conceput pentru a identifica sunetele produse de păsări. Fiecărui sunet i se atribuie o etichetă cu denumirea păsării care l-a produs. Modelul ML este antrenat folosind sunete etichetate și poate **prezice** eticheta (denumirea speciei de pasăre) care corespunde unor noi sunete. Datele sunt cel mai adesea etichetate de oameni, pentru ca modelele ML să se antreneze cu exemple etichetate corect.

Învățare prin consolidare

Învățarea prin consolidare este o abordare utilizată pentru **antrenarea modelelor** de **învățare automată (ML)**. Această abordare este folosită pentru a rezolva probleme cu un obiectiv clar, folosind atât recompense, cât și sancțiuni pentru a atinge obiectivul respectiv. Învățarea prin consolidare este utilizată la proiectarea mașinilor cu conducere autonomă sau pentru a juca jocuri complexe. De exemplu, un model de învățare prin consolidare ar putea fi folosit la proiectarea unei aplicații de șah. Modelul este antrenat să **prezică** mutările care maximizează recompensele și minimizează pierderile pentru a câștiga partida. În învățarea prin consolidare se folosesc recompense și sancțiuni pentru a identifica strategiile de atingere a unui obiectiv stabilit.

Învățare supravegheată

Învățarea supravegheată este o abordare utilizată pentru **antrenarea modelelor** de **învățare automată (ML)**. Aceste abordările folosesc cantități mari de **date etichetate** cu informații relevante de către operatori umani. Unul dintre tipurile de învățare supravegheată este **clasificarea**. Un exemplu de problemă de clasificare este identificarea tigrilor în sălbăticie. Datele constau într-un număr mare de imagini, cele cu tigri fiind etichetate ca atare.
Modelul ML este antrenat folosind imaginile etichetate pentru a **prezice** dacă într-o imagine anume există un tigru. Imaginile corect etichetate îi permit dezvoltatorului să știe în ce măsură predicțiile modelului sunt **corecte** și să ajusteze antrenarea modelului. După aceea, modelul ML poate fi folosit pentru a prezice dacă există un tigru în imagini complet noi. Pentru a produce predicții precise, abordările de tip învățare supravegheată depind de disponibilitatea unui volum suficient de date corect etichetate.

Învățare nesupravegheată

Învățarea nesupravegheată este o abordare utilizată pentru **antrenarea modelelor** de **învățare automată (ML)**. Dezvoltatorii ML antrenează modele de învățare nesupravegheată pentru a organiza **datele** pe baza asemănărilor. Acest proces are ca rezultat găsirea de tipare ascunse în volumul de date. Unul dintre tipurile de învățare nesupravegheată este gruparea (clustering). Un exemplu de

problemă de tip grupare este **a prezice** modul în care datele medicale pot fi grupate pentru a ajuta la diagnosticarea bolilor. Aceste grupuri sunt numite clustere și nu sunt cunoscute în prealabil. Modelul ML poate fi utilizat pentru a prezice dacă niște date medicale noi se încadrează într-unul dintre grupuri (clustere). Învățarea nesupravegheată poate fi utilă la rezolvarea problemelor în care oamenii nu știu ce să caute.

Acuratețe (ML)

Acuratețea se referă la cât de corect este ceva anume. În **învățarea automată (ML)**, acuratețea reprezintă un mod de a măsura cât de frecvent un **model ML** face o **predicție** corectă. De exemplu, un model de **clasificare** este conceput pentru a clasifica merele. Din 100 de imagini cu mere, 90 sunt clasificate corect. Aceasta înseamnă că modelul are o acuratețe de 90%. Acuratețea reprezintă una dintre modalitățile prin care sunt evaluate modelele ML. Cel mai adesea, acuratețea este utilizată împreună cu alți indicatori pentru a evalua calitatea unui model.

Caracter explicabil

Caracterul explicabil se referă la măsura în care ceva poate fi înțeles. În **învățarea automată (ML)**, explicarea sau caracterul explicabil ne ajută să înțelegem cum s-a ajuns la o anumită **predicție**. De exemplu, **modelele** cu **arbori de decizie** sunt transparente deoarece **nodurile** pot fi analizate într-un mod care poate fi înțeles. Majoritatea modelelor ML nu sunt pe deplin transparente, iar unele sunt mai transparente decât altele. Creșterea caracterului explicabil a unui model poate ajuta la remedierea problemelor și la combaterea **lipsei de imparțialitate**.

Grad de încredere

Gradul sau nivelul de încredere se referă la probabilitatea ca ceva anume să existe sau să aibă loc. În **învățarea automată (ML)**, gradul de încredere este o modalitate de a măsura nivelul de certitudine al unei **predicții**. Să luăm ca exemplu un **model** de **clasificare** conceput pentru a prezice dacă va ploua mâine. Modelul prezice cu un grad de încredere de 90% că mâine va ploua. Cu alte cuvinte, putem spune cu certitudine de 90% că mâine va ploua. Utilizarea acestui instrument (grad de încredere) pentru a evalua certitudinea predicțiilor ajută la evaluarea calității unui model ML.

Prag de încredere

Pragul de încredere este o valoare stabilită ca nivel minim acceptat de certitudine pentru **predicțiile unui model de învățare automată (ML)**. Pragul de încredere este stabilit de dezvoltatorul ML atunci când proiectează modelul ML. De exemplu, să presupunem că un model ML prezice cu un **grad de încredere** de 50% că mâine va fi viscol. Cu toate acestea, dacă pragul de încredere este stabilit la 60%, predicția respectivă va fi considerată incorectă. Cu alte cuvinte, dacă gradul de încredere al predicției nu este de cel puțin 60%, aceasta nu va fi acceptată drept **corectă**. Valoarea pragului de încredere se stabilește în funcție de natura problemei vizate. De exemplu, predicțiile ce țin de diagnosticarea afecțiunilor medicale necesită un prag de încredere mai ridicat decât recomandările de cântece din aplicațiile de muzică. Alegerea valorii pragului determină nivelul acceptabil de încredere în predicție.

Model ML

Un model de **învățare automată (ML)** este utilizat de o aplicație ML pentru a duce la capăt o sarcină sau pentru a rezolva o problemă. Modelul ML constituie o reprezentare a problemei care trebuie rezolvate. Dezvoltatorii ML folosesc volume mari de **date** reprezentative pentru o anumită problemă cu scopul de a **antrena** modelul să detecteze anumite tipare regulate. Rezultatul antrenamentului este un model care este folosit pentru a face **predicții** despre date noi în același context. De exemplu, mașinile cu conducere autonomă sunt construite folosind modele ML care anticipează când mașina trebuie să se oprească. Aceste modele sunt antrenate folosind milioane de exemple de situații în care mașinile trebuie să se oprească. Există multe tipuri diferite de modele, folosind diverse tipuri de date și diferite moduri de antrenare a modelelor. Toate modelele ML sunt antrenate pentru a detecta tipare regulate în **datele de antrenament** pentru a putea face astfel predicții despre date noi.

Fișă de model

O fișă de model de **învățare automată (ML)** este o modalitate de a consemna într-un mod structurat informații esențiale despre **modelele** ML. Fișele de model ML sunt scrise de dezvoltatorii ML atât pentru experți, cât și pentru non-experți. De exemplu, o aplicație ML este dezvoltată pentru a traduce în și din diferite limbi, cum ar fi din arabă în franceză și invers. O fișă de model include informații despre **acuratețea** traducerii realizate de model, precum și despre performanța acestuia în ceea ce privește diferitele jargoane, argouri și dialecte. Alte informații cuprinse în fișa de model ar putea include tipul de model ML, diferiți indicatori de performanță și chiar **prejudecăți**cunoscute. Fișele de model sunt create în timpul etapei explicative **ciclul de viață al proiectului AI**, pentru a oferi informații despre capacitățile și limitările modelului într-un mod ușor de înțeles.

Predicție

Modelele de **învățare automată (ML)** sunt **antrenate** să facă predicții. Predicția produsă de un model ML sugerează ce reprezintă **datele** sau ce ar putea fi util pentru îndeplinirea unei sarcini. De exemplu, un dezvoltator ML ar putea antrena un model pentru a prezice ce film și-ar putea dori cineva să vadă, pe baza obiceiurilor de vizionare ale persoanei respective. Modelul va genera o predicție după ce a fost instruit cu privire la filmele preferate ale unui număr mare de persoane. Principala sarcină a unui model ML este de a face predicții. Toate modelele ML fac predicții, chiar dacă în unele cazuri acestea nu sunt evidente pentru utilizator.

Prejudecată (non-imparțialitate)

Lipsa de imparțialitate (părtinirea sau prejudecata) se referă la o preferință pentru sau împotriva a ceva anume. De exemplu, un elev ar putea prefera orele de engleză în detrimentul celor de matematică și poate aloca mai mult timp temelor la engleză. Cu alte cuvinte, are o preferință față de limba engleză ca materie de studiu. Există multe tipuri de non-imparțialitate, printre care **discriminarea socială** și **datele selective**. Dezvoltatorii de software de **învățare automată (ML)** trebuie să analizeze cu atenție dacă **datele** pe care le folosesc pentru a **antrena modelele** ML sunt părtinitoare sau nu. Lipsa de imparțialitate poate avea ca rezultat avantajarea unei persoane, a unui grup sau a anumitor idei sau credințe în dauna altora.

Prejudecăți în date

Prejudecățile din date se referă la **lipsa imparțialității**, așa cum se reflectă aceasta în **datele** utilizate pentru a **antrena modele** de **învățare automată (ML)**. Aceasta poate duce la antrenarea modelelor ML astfel încât să genereze **predicții** părtinitoare. De exemplu, unele modele de recunoaștere facială sunt părtinitoare față de chipurile cu anumite culori ale pielii deoarece au fost antrenate folosind în mare parte imagini de fețe cu o singură culoare a pielii. Există mai multe posibile surse potențiale de non-imparțialitate în ceea ce privește datele. Printre acestea se numără datele incomplete și datele care reflectă **prejudecăți de natură socială**. Detectarea prezenței unei lipse de imparțialitate a datelor este importantă pentru a evita ca modelele ML să genereze predicții părtinitoare.

Prejudecată socială

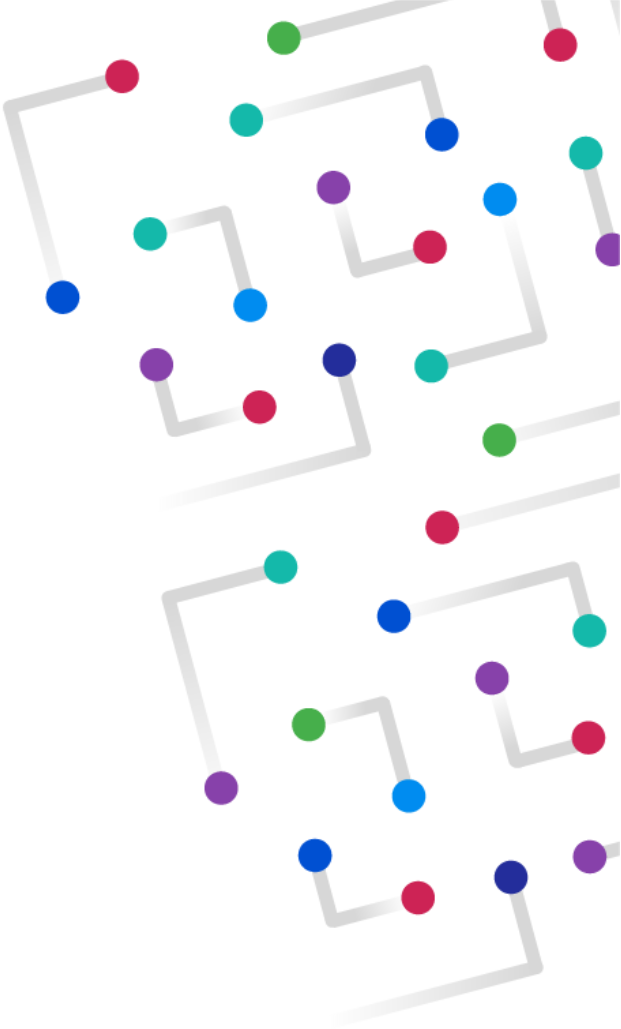
Prejudecățile sociale sunt convingeri împărtășite de grupuri mari de oameni sau chiar de societate în ansamblul acesteia. Există multe tipuri diferite, cum ar fi prejudecățile rasiale, de gen sau etnice. Un exemplu de prejudecată de gen este ideea că o carieră în domeniul ingineriei li se potrivește mai mult bărbaților decât femeilor. **Datele** colectate de la grupuri mari de oameni ar putea reflecta astfel de prejudecăți sociale, ducând astfel la **prejudecăți în date**. Dacă datele care reflectă prejudecăți sociale sunt folosite pentru a **antrena modele** de **învățare automată (ML)**, acest lucru poate duce la modele care generează **predicții** părtinitoare. În ML, este important ca prejudecățile sociale reflectate în **datele de antrenament** să fie eliminate, pentru a evita obținerea de rezultate discriminatorii sau nedrepte.

Rule-based (bazat pe reguli)

Abordarea bazată pe reguli reprezintă o modalitate de proiectare a sistemelor folosind un set de reguli predefinite. De exemplu, o aplicație software de jucat X și 0 este proiectată utilizând reguli care arată cum trebuie să joci pentru a câștiga. Regulile sunt definite de oameni care sunt de obicei experți în domeniul problemei respective. Sistemele de **inteligență artificială (AI)** construite folosind o abordare bazată pe reguli sunt cunoscute și sub denumirea de „AI de modă veche”. Sistemele bazate pe reguli contrastează cu sistemele **„data-driven” (bazate pe date)**, unde **datele** sunt folosite ca exemple de rezolvare a problemei. Sistemele bazate pe reguli sunt utile pentru rezolvarea problemelor în care pot fi generate și respectate reguli care acoperă majoritatea situațiilor.

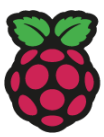
Această resursă este licențiată de către Raspberry Pi Foundation sub o licență publică internațională Creative Commons Attribution-NonComercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0). Pentru mai multe informații despre această licență, accesează creativecommons.org/licenses/by-nc-nd/4.0.

Experience AI



Acest glosar a fost creat de Raspberry Pi Foundation în cadrul programului educațional Experience AI. Află mai multe la **www.experience-ai.org**

Această resursă este licențiată de către Raspberry Pi Foundation sub o licență publică internațională Creative Commons Attribution-NonComercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0). Pentru mai multe informații despre această licență, accesează creativecommons.org/licenses/by-nc-nd/4.0.



Raspberry Pi
Foundation