

Ministerul Educației

Programă școlară pentru disciplina opțională

**Introducere în învățarea automată ca ramură a inteligenței
artificiale**

[Curriculum la decizia școlii pentru liceu]

Aprobată prin ordinul ministrului
nr.din

București, 2022

NOTA DE PREZENTARE

Programa școlară „**Introducere în învățarea automată ca ramură a inteligenței artificiale**” reprezintă o ofertă curriculară de disciplină opțională pentru învățământul liceal, proiectată pentru un buget de timp de 1 oră/săptămână, pe durata unui an școlar.

Disciplina se poate studia în clasa a XI-a sau a XII-a și necesită competențe minime de programare. Prin urmare, această disciplină opțională se adresează cu precădere elevilor de la filiera teoretică, profil real.

Studiul disciplinei „**Introducere în învățarea automată ca ramură a inteligenței artificiale**” se va desfășura cu întreg colectivul de elevi și necesită acces la dispozitive de calcul.

Învățarea automată (*machine learning*) este, în prezent, principala abordare de a construi sisteme inteligente, adică sisteme ce pot rezolva sarcini complexe cum ar fi recunoașterea unei persoane într-o imagine, traducerea unui text dintr-o limbă străină sau abilitatea de a juca șah la nivel profesionist și exemplele pot continua. Crearea de astfel de sisteme inteligente reprezintă obiectul de studiu al inteligenței artificiale, domeniu larg de cercetare aflat la intersecția dintre: informatică, matematică, inginerie, statistică, neuroștiințe, științele educației etc. **Învățarea automată este, așadar, o ramură a inteligenței artificiale.**

Învățarea automată se bazează pe algoritmi care permit sistemelor să învețe din experiență. În timp ce în sistemele clasice sunt programați explicit toți pașii de rezolvare a problemei, sistemele bazate pe învățare automată primesc multe exemple și învață singure să obțină soluția potrivită.

Programa școlară a disciplinei „**Introducere în învățarea automată ca ramură a inteligenței artificiale**” răspunde nevoilor de continuare a eforturilor de alfabetizare digitală din perspectiva noilor cerințe socio-profesionale, printr-un curriculum relevant, vizând formarea unui set de competențe digitale, pe care absolvenții învățământului liceal să le poată valorifica în viața activă.

În elaborarea acestei programe școlare, a fost luată în considerare **Cartea albă privind inteligența artificială – O abordare europeană axată pe excelență și încredere**. Aceasta a fost publicată la 19 februarie 2020 de către Comisia Europeană și prezintă principiile de bază ale unui viitor cadru de reglementare al UE pentru inteligența artificială în Europa:

https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/excellence-trust-artificial-intelligence_ro

De asemenea, s-a ținut cont de preocupările intense la nivel național, precum:

- elaborarea cadrului strategic național în domeniul inteligenței artificiale, parte a proiectului *Cadru strategic pentru adoptarea și utilizarea de tehnologii inovative în administrația publică 2021 – 2027 – soluții pentru eficientizarea activității*:

<https://www.adr.gov.ro/cadru-strategic-pentru-adoptarea-si-utilizarea-de-tehnologii-inovative-in-administratia-publica-2021-2027-solutii-pentru-eficientizarea-activitatii-cod-sipoca-704/>

- *Strategia privind digitalizarea educației din România*:

<https://www.edu.ro/sites/default/files/SMART.Edu%20-%20document%20consultare.pdf>

- *Repere pentru proiectarea, realizarea și evaluarea curriculumului național* (Anexa la OME 3239/05.02.2021)

- recomandarea ANIS (*Asociația Patronală a Industriei de Software și Servicii*), care atrage atenția asupra includerii în *Strategia Națională pentru Inteligență Artificială* a unor măsuri concrete prin care **curriculumul școlar să fie actualizat și adaptat evoluției tehnologice**

<http://ina.gov.ro/tema-lunii-digitalizarea-si-inteligenta-artificiala-in-administratia-publica/>

Pentru dezvoltarea unor soluții informatice care au în vedere învățarea automată (machine learning) se recomandă utilizarea limbajului de programare *Python*. Alegerea acestui limbaj a fost motivată de posibilitatea accesării gratuite a unor biblioteci care oferă mecanisme/facilități de utilizare a algoritmilor specifici învățării automate. Mai mult, în momentul de față, atât în industria software cât și în mediul academic, implementarea algoritmilor de învățare automată (machine learning) se face frecvent în limbajul *Python*.

Din perspectiva modelului de proiectare curriculară, programa școlară de față include următoarele elemente:

- **Notă de prezentare**
- **Competențe generale**
- **Competențe specifice și exemple de activități de învățare**
- **Conținuturi**
- **Sugestii metodologice**
- **Referințe bibliografice**

Competențele sunt ansambluri structurate de cunoștințe, abilități și atitudini formate prin învățare, care permit rezolvarea unor probleme specifice unui domeniu sau a unor probleme generale, în contexte particulare diverse.

Competențele generale vizează achizițiile de cunoaștere și de comportament ale elevului din învățământul liceal.

Competențele specifice sunt derivate din competențele generale, reprezintă etape în dobândirea acestora, formându-se pe durata unui an școlar. Pentru realizarea competențelor specifice, în programă sunt propuse exemple de activități de învățare, cu caracter opțional, care valorifică experiența concretă a elevului și care integrează strategii didactice adecvate unor contexte de învățare variate.

Conținuturile învățării reprezintă baza de operare pentru formarea competențelor.

Sugestiile metodologice includ strategii didactice și au rolul de a orienta cadrul didactic în aplicarea programei. Pentru a facilita demersul cadrului didactic, sugestiile metodologice includ, de asemenea, exemple de activități de învățare / evaluare, precum și referința la un suport de curs complet și alte resurse utile.

Referințele bibliografice includ o listă de lucrări relevante pentru domeniul de studiu respectiv, care sprijină cadrul didactic în aplicarea programei școlare.

COMPETENȚE GENERALE

1. Implementarea algoritmilor în limbajul Python
2. Elaborarea creativă de soluții informatice care vizează aspecte sociale, culturale și personale, utilizând algoritmi care au la bază învățarea automată

COMPETENȚE SPECIFICE ȘI EXEMPLE DE ACTIVITĂȚI DE ÎNVĂȚARE

1. Implementarea algoritmilor în limbajul Python

1.1 Utilizarea unui mediu de dezvoltare pentru limbajul Python

- folosirea interpretorului de comenzi din mediul de dezvoltare utilizat pentru executarea unor scripturi Python model, modificarea lor conform unor cerințe și crearea unor secvențe noi de cod
- familiarizarea cu funcționalitățile mediului de dezvoltare pentru editarea codului cum ar fi indentarea codului, sugestii de completare, ajutor contextual
- familiarizarea cu funcționalitățile mediului de dezvoltare pentru executarea codului cum ar fi detectarea erorilor sau depanarea secvențelor de cod

1.2 Elaborarea unor scripturi Python care prelucrează tipuri simple de date

- recunoașterea rolului unor elemente prezente în sintaxa instrucțiunilor limbajului Python prin analogia dintre o secvență de instrucțiuni Python și secvența corespunzătoare scrisă într-un limbaj de programare studiat sau în pseudocod
- modificarea unor secvențe de instrucțiuni pentru a pune în evidență rolul elementelor componente (de exemplu parcurgerea valorilor naturale dintr-un interval în ordine crescătoare/descrescătoare, cu un anumit pas)
- exersarea instrucțiunilor de intrare-ieșire
- rezolvarea unor probleme ce utilizează tipurile de date simple și operatorii specifici limbajului Python

1.3 Prelucrarea datelor structurate

- prezentarea datelor structurate (tablouri, liste, dicționare, tupluri) prin comparație cu alte limbaje de programare studiate și prin exemple concrete cum ar fi: folosirea unui dicționar pentru a memora o agendă telefonică, a unui tablou pentru înregistrarea măsurătorilor unui senzor, memorarea dimensiunilor unei imagini într-un tuplu sau stocarea numelor elevilor înscriși la un concurs într-o listă
- crearea, inițializarea, conversia și utilizarea unor date structurate, de exemplu pentru a ordona date sau a căuta valori

1.4 Definirea și apelarea funcțiilor în Python

- prezentarea structurii unei funcții în Python (antet, parametri, mecanisme de transmitere a parametrilor și returnare a valorilor)
- utilizarea unor funcții predefinite, de exemplu, funcția **len** ce returnează lungimea unei structuri de date, funcția **str** pentru conversie în șir de caractere
- rezolvarea de probleme folosind funcții definite de utilizator (de exemplu, c.m.m.d.c. a două numere sau suma cifrelor unui număr)
- analizarea avantajelor utilizării funcțiilor: reutilizarea codului, elaborarea algoritmilor prin descompunerea problemei în subprobleme mai simple, reducerea numărului de erori care pot apărea la scrierea programelor, depistarea cu ușurință a erorilor, realizarea unor programe ușor de urmărit (lizibile).
- rezolvarea de probleme care necesită date structurate (de exemplu, listele sau dicționarele ce pot fi utilizate pentru gestiunea cărților dintr-o bibliotecă sau pentru gestiunea unei agende telefonice, evidențiind funcțiile predefinite, specifice fiecărui tip de date structurate)

1.5 Utilizarea bibliotecilor în elaborarea unor scripturi

- prezentarea și exemplificarea utilizării unor funcții uzuale din cadrul unor biblioteci
 - o Matplotlib - grafice de funcții și ale derivatelor lor (de exemplu, funcția cosinus și derivata acesteia – funcția sinus)
 - o Numpy - lucrul cu date multidimensionale: filtrarea datelor după anumite criterii, calculul de funcții matematice (medie, minim, maxim) pe anumite dimensiuni ale datelor
 - o PIL (Python Image Library) – conversia în tonuri de gri, binarizarea imaginilor, filtrarea imaginilor
- realizarea de aplicații simple ce folosesc funcțiile exemplificate cum ar fi: trasarea graficului nivelului de poluare, calculul unor valori medii înregistrate pentru diverse tipuri de particule ce poluează sau compresia unor imagini relevante pentru poluare.

2. Elaborarea creativă de soluții informatice care vizează aspecte sociale, culturale și personale, utilizând algoritmi care au la bază învățarea automată

2.1 Caracterizarea inteligenței artificiale în contextul aplicațiilor ei în viața cotidiană

- prezentarea în evoluție temporală a sistemelor inteligente cu exemple renumite – Deep Blue, AlexNet, AlphaGo sau AlphaFold
- identificarea aplicațiilor inteligenței artificiale în viața cotidiană (funcționarea mașinilor autonome, recunoașterea facială și vocală)
- exersarea reflecției critice din perspectiva utilizării inteligenței artificiale în viața de zi cu zi
- vizualizarea unor materiale video ce redau exemple de succes privind utilizarea inteligenței artificiale
- prezentarea învățării automate (machine learning) ca ramură a inteligenței artificiale

2.2 Caracterizarea învățării automate în comparație cu programarea clasică

- recunoașterea situațiilor în care învățarea automată este utilă sau chiar necesară
- definirea problemei în învățarea automată: mulțimea de date pe care se bazează învățarea (dataset) și formularea obiectivului/ obiectivelor învățării automate
- exemplificarea unor situații concrete de utilizare a învățării automate (de exemplu, identificarea unor obiecte în imagini, predicția unor valori pe baza istoricului)
- identificarea diferențelor dintre programarea clasică și învățarea automată
- exemplificarea de probleme abordate clasic versus prin învățare automată (de exemplu, identificarea unei categorii de obiecte în imagini)
- prezentarea unor probleme care au soluții dificil de implementat prin programarea clasică din cauza numărului mare de cazuri posibile

2.3 Proiectarea contextului de utilizare și implementarea unor algoritmi specifici învățării automate

- formularea problemelor principale în învățarea automată (clusterizare, clasificare, regresie)
- identificarea asemănărilor și deosebirilor între clusterizare, clasificare și regresie
- proiectarea, pregătirea și implementarea de algoritmi pentru rezolvarea acestor probleme (de exemplu Kmeans pentru clusterizare, KNN pentru clasificare și regresie liniară)
- utilizarea funcției FORECAST.LINEAR în Excel, ca suport de vizualizare a rezultatelor regresiei liniare

2.4 Alegerea unor algoritmi eficienți de învățare automată pentru rezolvarea problemelor

- analizarea situațiilor în care predicția rezultată în urma rulării unui algoritm nu este conformă cu realitatea (de exemplu, utilizarea unor seturi de date neechilibrate care generează predicții părtinitoare)
- punerea în discuție a modului în care elevii au rezolvat o problemă și a unor situații noi în care ar putea fi folosit același algoritm
- alegerea algoritmului de învățare automată (supervizat sau nesupervizat) în funcție de tipul de date disponibile: etichetate sau neetichetate, continue sau discrete
- conștientizarea achizițiilor de învățare personală în urma aplicării algoritmilor specifici învățării automate (Ce folosesc pentru viață? Ce am învățat din fiecare situație problemă-clusterizare, clasificare, regresie – și va fi folositor mâine? Ce pot explica altei persoane? etc.)

CONȚINUTURI

Domenii de conținut	Conținuturi
Elemente introductive în învățarea automată ca subdomeniu al inteligenței artificiale	<ul style="list-style-type: none"> - Repere importante din istoricul inteligenței artificiale și al învățării automate - Modul de proiectare și operare a unui sistem inteligent bazat pe învățare automată - Tipuri de seturi de date utilizate în învățarea automată: etichetate, neetichetate, de antrenare și de test - Atributele unui sistem inteligent bazat pe învățare automată: autonomie, robustețe, capacitate de adaptare, învățare din experiență - Rolul inteligenței artificiale bazate pe învățarea automată în societate - Elemente de etică în utilizarea inteligenței artificiale bazate pe învățarea automată (protejarea datelor personale, bias, diferența între corelație și cauzalitate)

Domenii de conținut	Conținuturi
Medii de dezvoltare Python	<ul style="list-style-type: none"> - Elemente de lucru în diverse medii de dezvoltare pentru limbajul Python: interpretor de comenzi, fișier de comenzi, script, linie de comandă - Structura generală a unui script Python (biblioteci, variabile, instrucțiuni, funcții) - Modul de executare și depanare a unui script Python
Elemente de bază ale limbajului de programare Python	<ul style="list-style-type: none"> - Elemente de vocabular al limbajului Python (identificatori, cuvinte cheie, constante, separatori, operatori, comentarii) - Tipuri de date de bază în Python (date numerice: int, float; șiruri de caractere; date logice: bool) - Funcții pentru citirea și afișarea datelor (input, print cu sau fără formatare) - Instrucțiuni: expresie, alternativă – if, și repetitive – for, while - Tipuri structurate de date (liste, tupluri, dicționare): inițializare, operații specifice (adăugare/ ștergere elemente, parcurgeri) - Funcții definite de utilizator (antet, parametri, domeniul de vizibilitate al variabilelor, mecanisme de transmitere a parametrilor și returnare a valorilor, apel)
Biblioteci Python utilizate în învățarea automată	<ul style="list-style-type: none"> - <i>Numpy</i> - bibliotecă de calcul numeric: tipuri de date multidimensionale și funcții predefinite (min, max, sort, len, shape, type) - <i>Matplotlib</i> – bibliotecă pentru vizualizarea graficelor de funcții matematice: funcții predefinite (plot, show) - <i>PIL (Python Image Library)</i> – bibliotecă pentru manipularea și prelucrarea imaginilor: funcții predefinite (open, save, convert, filter, pad)
Învățare automată nesupervizată	<ul style="list-style-type: none"> - Definiția și caracteristicile învățării automate nesupervizate (utilizează seturi de date neetichetate, descoperă structura intrinsecă a datelor) - <i>Clusterizarea</i> ca tehnică de învățare nesupervizată: obiectivul învățării, criteriile de grupare a datelor pe baza similarității - Algoritmii de clusterizare <i>K-means</i>: set de date neetichetate (convexe și neconvexe), număr de clustere, funcția de similaritate, corespondența între date și clustere ca date de ieșire - Criterii de interpretare a rezultatelor obținute prin clusterizare: puritatea și densitatea clusterelor - Avantajele utilizării algoritmului de clusterizare <i>K-means</i>: convergență rapidă, aplicabilitate pe scară largă, ușurința implementării - Dezavantajele utilizării algoritmului de clusterizare <i>K-means</i>: dificultatea alegerii numărului optim de clustere, funcționarea defectuoasă pe seturi de date neconvexe
Învățare automată supervizată	<ul style="list-style-type: none"> - Definiția și caracteristicile învățării automate supervizate (utilizează seturi de date etichetate de antrenare și de test, descoperă corespondența între date de intrare și etichete) - <i>Clasificarea</i> ca tehnică de învățare supervizată: obiectivul învățării, natura etichetelor (discrete, fără relație de ordine) - Algoritmii de clasificare <i>KNN</i>: set de date etichetate, număr de etichete, distanța între date (vecini), numărul celor mai apropiați vecini considerați, corespondența între date și etichete ca date de ieșire - Criterii de interpretare a rezultatelor obținute prin clasificare: acuratețe pe datele de antrenare și de test, matricea de confuzie

Domenii de conținut	Conținuturi
	<ul style="list-style-type: none"> - Regresia ca tehnică de învățare supervizată: obiectivul învățării, natura etichetelor (continue, cu relație de ordine) - Algoritmii de regresie liniară: dependența liniară dintre datele de ieșire (etichete) și datele de intrare, formule de calcul pentru parametrii funcției liniare - Criterii de interpretare a rezultatelor obținute prin regresie: eroarea medie - Criterii de analiză comparativă între clasificare și regresie (rezultatele vizate sunt discrete sau continue, existența unei relații de ordine între rezultatele vizate)

SUGESTII METODOLOGICE

Fiecare profesor proiectează unitățile de învățare, stabilește succesiunea logică de parcurgere a acestora și bugetul de timp alocat, într-o manieră flexibilă, având în vedere nivelul de achiziții și interesele elevilor, precum și punând în valoare experiența și creativitatea acestora.

Este recomandată abordarea, în egală măsură, a aspectelor teoretice și practice, astfel încât, la finalul cursului, elevii să înțeleagă principiile de bază utilizate în învățarea automată.

Parcurgerea cu succes a acestei programe este facilitată de *cunoștințe și abilități de programare* folosind unul dintre limbajele studiate la școală. De asemenea, noțiuni de *matematică* sunt necesare ca suport teoretic pentru a înțelege anumite concepte de învățare automată.

Conținuturile sunt structurate în două părți. Prima parte vizează familiarizarea cu limbajul *Python*, iar în partea a doua, sunt prezentate concepte de bază de învățare automată, algoritmi fiind implementați în limbajul *Python*.

Pentru implementarea de secvențe de cod în limbajul *Python* pot fi folosite platformele <https://pythonisti.ro>, <https://trinket.io/python>, *Google Colab* sau mediile de dezvoltare *Python 3.0 IDE* sau *Visual Studio Code*. Se recomandă folosirea platformei *Google Colab* deoarece nu necesită nicio configurare prealabilă, fiind suficient doar un cont *Gmail*, și permite crearea și partajarea de documente care conțin: secvențe de cod care rulează în timp real, reprezentări vizuale și text. În plus, utilizarea *Google Colab* permite accesul gratuit la resurse de calcul paralel (acceleratoare grafice) din *Google Cloud*.

În procesul de predare-învățare-evaluare, activitatea va fi orientată pe analiza unor situații cotidiene și pe identificarea unor soluții practice ce folosesc algoritmi de învățare automată (*machine learning*).

Activități de învățare

Este recomandat ca prezentarea conceptelor și algoritmilor de învățare automată să fie însoțite de exemple practice de utilizare din viața de zi cu zi și/sau de materiale video relevante.

Prezentarea conținuturilor poate debuta cu analiza unei secvențe de cod relevantă și modificarea acesteia pentru a obține diverse rezultate.

Elevii pot lucra individual sau în echipă pentru aplicarea algoritmilor studiați pe diverse seturi de date. De exemplu, referitor la strategia de *învățare supervizată*, problema clasificării poate fi prezentată utilizând setul de date *Iris*, iar elevii își pot consolida cunoștințele aplicând algoritmul de clasificare pe alt set de date cum ar *mnist*.

Elevii pot organiza discuții și dezbateri plecând de la experiențele practice furnizate de profesor sau disponibile pe Internet, pentru a înțelege cum pot fi utilizați algoritmi de învățare automată în viața de zi cu zi.

Activități de feedback și evaluare

Se recomandă lucrul pe proiecte, individual sau în echipă, cu teme practice inspirate din viața cotidiană, care să stimuleze creativitatea elevilor și să le faciliteze aplicarea cunoștințelor în diverse domenii.

Prezentarea proiectelor oferă o ocazie de reflecție asupra propriei experiențe și creează canale de comunicare între elevi și profesor.

În ceea ce privește etica în utilizarea inteligenței artificiale pot fi organizate dezbateri, jurnale de reflecție, plecând de la întrebări precum:

- Datele personale sunt protejate? Viața privată este respectată?
- Care ar fi consecințele folosirii unui set de date inadecvat?
- În ce contexte poate apărea confuzia corelație și cauzalitate?

Resurse

- **Python & Machine Learning pentru liceu - Bine ați venit! (pyml.ro)**
 - Platformă online, cu acces liber, ce detaliază conținuturile științifice, inclusiv exemple practice și oferă acces la un interpretor Python pentru editarea, executarea și depanarea scripturilor, precum și acces la acceleratoare grafice pe mașini virtuale prin intermediul platformei Google Colab
- **Curs de Python 3 pentru începători - Online interactiv (pythonisti.ro)**
 - Platformă de învățare online, cu acces liber, pentru dezvoltarea abilităților de programare în limbajul Python
- **Jocuri didactice ce au ca scop inițierea în Python**
 - <https://silentteacher.toxicode.fr/>
 - <https://compute-it.toxicode.fr/>
- **Mini-tutoriale video online pentru inițierea în Python (seria Python is fun! – canalul youtube Infogim)**
 - https://www.youtube.com/watch?v=c7mD2_v5-0Q&t=2s
 - https://www.youtube.com/watch?v=Fq6UvEf_n3A&t=2s
 - https://www.youtube.com/watch?v=X18gx_OYTt4&t=46s
- **<https://trinket.io/python>**
 - Platformă online, cu acces liber, ce permite dezvoltarea de scripturi Python și furnizează exemple de proiecte
- **<https://ml.infobits.ro/machine-learning-aplicatii-practice-online.php>**
 - Platformă online, cu acces liber dedicată aplicațiilor practice interactive de învățare automată

REFERINȚE BIBLIOGRAFICE

- Elements of AI <https://course.elementsofai.com/ro/>
- AI for Everyone, Andrew Ng, Coursera, <https://www.coursera.org/learn/ai-for-everyone>
- Deep Learning (partea I), de Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press (2017) <https://www.deeplearningbook.org/>
- Pattern Recognition and Machine Learning (partea a III-a și partea a IV-a), de Christopher Bishop, Springer (2006), <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf>
- AI/ML/Big Data News: <https://allainews.com/>

Autori

Prof. dr. Ana-Maria Arișanu, Colegiul Național "Mircea cel Bătrân", Rm. Vâlcea

Prof. gr. II Antonia Haller, Colegiul Național "Mircea cel Bătrân", Rm. Vâlcea

Prof. gr. I Emil Onea, Colegiul Național "Unirea", Focșani, UPIR

Dr. cercetător Răzvan Pașcanu, DeepMind Londra, EEML

Dr. ing. cercetător Viorica Pătrăucean, DeepMind Londra, EEML

Cu sprijin din partea:

<https://www.eeml.eu/> Comunitatea Eastern European Machine Learning

<http://aria-romania.org/> Asociația Română pentru Inteligență Artificială

<http://www.upir.ro/> Uniunea Profesorilor de Informatică din România

<https://www.airomania.eu> Comunitatea AIRomania