# FIȘA DISCIPLINEI

**1. Date despre program**

| | |
|---|---|
| 1.1 Instituția de învățământ superior | Universitatea Tehnică din Cluj-Napoca |
| 1.2 Facultatea | Facultatea de Științe (Centrul Universitar Nord din Baia Mare) |
| 1.3 Departamentul | Departamentul de Matematică și Informatică |
| 1.4 Domeniul de studii | Informatică |
| 1.5 Ciclul de studii | Master |
| 1.6 Programul de studii / Calificarea | Informatică și inginerie software (Baia Mare, limba română) / Master profesional în domeniul Informatică |
| 1.7 Forma de învățământ | IF – învățământ cu frecvență |

**2. Date despre disciplină**

| | | | | | |
|---|---|---|---|---|---|
| 2.1 Denumirea disciplinei | Învățare Profundă Aplicativă cu Python | | | Codul disciplinei | 2.00 |
| 2.2 Titularul de curs | *Conf. univ. dr. Cosmin Nicolae SABO – cosmin.sabo@mi.utcluj.ro* | | | | |
| 2.3 Titularul activităților de seminar / laborator / proiect / practică | *Conf. univ. dr. Cosmin Nicolae SABO – cosmin.sabo@mi.utcluj.ro* | | | | |
| 2.4 Anul de studiu | I | 2.5 Semestrul | 1 | 2.6 Tipul de evaluare | E |
| 2.7 Regimul disciplinei | Categoria formativă | | | | DF |
| | Opționalitate | | | | DOB |

**3. Timpul total estimat**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3.1 Număr de ore pe săptămână | 4 | din care: | 3.2 Curs | 2 | 3.3 Seminar | - | 3.3 Laborator | 2 | 3.3 Proiect | - | 3.3 Practică | - |
| 3.4 Număr de ore pe semestru | 56 | din care: | 3.5 Curs | 28 | 3.6 Seminar | - | 3.6 Laborator | 28 | 3.6 Proiect | - | 3.3 Practică | - |

| | |
|---|---|
| 3.7 Distribuția fondului de timp (ore pe semestru) pentru studiu individual și evaluare: | |
| (a) Evaluare | 8 |
| (b) Studiul după manual, suport de curs, bibliografie și notițe | 34 |
| (c) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren | 16 |
| (d) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri | 76 |
| (e) Tutoriat | 4 |
| (f) Alte activități | 6 |
| 3.8 Total ore studiu individual și evaluare (suma (3.7(a)...3.7(f)) | 144 |
| 3.9 Total ore pe semestru (3.4+3.8) | 200 |
| 3.10 Numărul de credite | 8 |

**4. Precondiții** (acolo unde este cazul)

| | |
|---|---|
| 4.1 de curriculum | Nu există precondiții obligatorii. Sunt utile noțiuni de programare, algebră liniară și probabilități din ciclul de licență; ele sunt reactualizate prin materialele de nivelare ale disciplinei (săptămânile 1-4). |
| 4.2 de competențe | Programare de bază într-un limbaj oarecare (variabile, structuri de control, funcții); utilizarea calculatorului și a unui browser; citirea documentației tehnice în limba engleză. Nu sunt necesare cunoștințe anterioare de învățare automată. |

**5. Condiții** (acolo unde este cazul)

| | |
|---|---|
| 5.1. de desfășurare a cursului | Sală de curs cu videoproiector și acces la internet; prelegerile includ demonstrații live (caiete de lucru, modele antrenate în timpul cursului), pe care studenții le pot urmări pe laptopul personal.<br>Materialele de curs, chestionarele, sondajele și anunțurile se publică pe platforma Campus Virtual și pe canalele de comunicare instituționale ale disciplinei. |
| 5.2. de desfășurare a seminarului / laboratorului / proiectului | Laborator cu calculatoare conectate la internet, cu Python (versiune curentă) și bibliotecile necesare instalate; acces la un mediu de calcul în cloud cu GPU, disponibil gratuit studenților, și la o platformă de versionare a codului; se recomandă lucrul pe laptopul personal. Nu este necesar un GPU local: sarcinile de bază rulează pe CPU.<br>Conturi necesare din săptămâna 1: Campus Virtual, contul instituțional Microsoft 365 și conturile gratuite pentru mediul de calcul în cloud și pentru platforma de versionare a codului, indicate la primul laborator.<br>Activitatea se desfășoară individual, în perechi și în echipe de proiect de 3 studenți, mixte după programul de licență absolvit, constituite după prima săptămână; livrabilele se predau prin Campus Virtual până la termenele anunțate. Asistenții AI sunt permiși la laborator și la proiect, cu declararea lor în fiecare livrabil, și interziși la examen.<br>Prezența la laborator este obligatorie, conform regulamentului UTCN privind activitatea profesională a studenților. |

**6. Competențele specifice acumulate**

| | |
|---|---|
| Competențe profesionale | C2. Modelarea matematică folosind cercetarea științifică actuală: formularea unei probleme reale de clasificare, regresie, prognoză sau detecție de anomalii ca problemă de învățare supervizată sau nesupervizată și alegerea arhitecturii de rețea neuronală adecvate tipului de date (tabulare, imagini, serii de timp, text).<br>C3. Capacitatea de operare cu algoritmi și de aplicare a lor practică: implementarea în Python, cu un framework modern de învățare profundă, antrenarea, optimizarea și evaluarea rețelelor neuronale dense, convoluționale, recurente și de tip transformer, inclusiv prin învățare prin transfer cu modele preantrenate.<br>C4. Identificarea, definirea și dezvoltarea unei teme de cercetare: proiectarea și realizarea unui studiu aplicativ complet și reproductibil (date, referință clasică, model, analiză de erori), cu compararea onestă a rezultatelor.<br>C5. Redactarea și susținerea unei lucrări de cercetare: documentarea studiului într-un raport în format de articol scurt, cu demonstrație publică a modelului și susținere în fața colegilor. |
| Competențe transversale | CT1. Desfășurarea unei activități responsabile, riguroase și cu respectarea normelor deontologiei profesionale: metodologie experimentală corectă (fără scurgeri de date), respectarea licențelor seturilor de date și a protecției datelor personale, declararea utilizării asistenților AI.<br>CT2. Desfășurarea eficientă și eficace a activităților organizate în echipă: roluri asumate și rotite în echipe mixte, planificarea etapelor proiectului, evaluare între colegi.<br>CT3. Utilizarea eficientă a surselor informaționale și a resurselor de comunicare și formare profesională asistată, în limba română și în limba engleză: documentația bibliotecilor utilizate, manuale deschise, cursuri online, comunități de practică. |

**7. Rezultatele așteptate ale învățării**

| | |
|---|---|
| Cunoștințe | RI1. Explică principiile învățării supervizate și ale rețelelor neuronale: funcții de pierdere, gradient și backpropagation, algoritmi de optimizare, regularizare, precum și limitele lor teoretice și practice.<br>RI2. Descrie critic arhitecturile fundamentale (perceptron multistrat, rețele convoluționale, rețele recurente, transformer, autoencoder) și condițiile în care fiecare este adecvată unui tip de date și unei probleme.<br>RI3. Cunoaște metodologia experimentală a învățării automate (împărțirea datelor, metrici, referințe clasice, reproductibilitate, urmărirea experimentelor) și cerințele etice și legale de bază ale aplicațiilor cu învățare profundă (bias, protecția datelor, reglementările europene privind inteligența artificială). |
| Abilități | RI4. Implementează, cu un framework Python de învățare profundă, pipeline-uri complete de date, model, antrenare și evaluare, rulate local sau în medii de calcul în cloud.<br>RI5. Aplică învățarea prin transfer cu modele preantrenate de viziune și de text din depozite publice de modele și le adaptează la seturi de date proprii.<br>RI6. Diagnostichează antrenarea (supra-antrenare, sub-antrenare, instabilitate) din curbele de învățare și alege intervențiile potrivite (date, arhitectură, regularizare, hiperparametri).<br>RI7. Compară experimental modele de învățare profundă cu metode clasice, analizează erorile și argumentează alegerea într-un raport scurt în format științific, cu o demonstrație a modelului livrată ca aplicație. |
| Responsabilitate și autonomie | RI8. Conduce, într-o echipă mixtă, un proiect aplicativ de învățare profundă de la formularea problemei la livrarea unei demonstrații, asumându-și un rol, termenele și contribuția proprie, vizibilă în istoricul codului.<br>RI9. Evaluează onest calitatea și riscurile propriilor modele (bias, scurgeri de date, generalizare, costuri de calcul) și declară transparent instrumentele folosite, inclusiv asistenții AI. |

**8. Obiectivele disciplinei** (reieșind din grila competențelor specifice acumulate)

| | |
|---|---|
| 8.1 Obiectivul general al disciplinei | Formarea capacității de a rezolva probleme reale cu învățare profundă în Python, de la date la livrare, cu rigoare experimentală și responsabilitate, într-un mod accesibil absolvenților de licență din Informatică, Matematică și Științe Economice. |
| 8.2 Obiectivele specifice | O1. Înțelegerea mecanismelor de bază ale rețelelor neuronale (pierdere, gradient, backpropagation, optimizare, regularizare).<br>O2. Stăpânirea unui framework Python modern de învățare profundă pentru construirea, antrenarea, diagnosticarea și evaluarea modelelor.<br>O3. Aplicarea arhitecturilor potrivite pentru date tabulare, imagini, serii de timp și text.<br>O4. Folosirea învățării prin transfer și a modelelor preantrenate din depozite publice.<br>O5. Evaluarea și compararea onestă a modelelor de învățare profundă cu metode clasice, cu analiza erorilor.<br>O6. Livrarea reproductibilă a unui model ca aplicație demonstrativă, cu cod versionat.<br>O7. Lucrul în echipă mixtă și comunicarea rezultatelor într-un raport în format științific, cu utilizarea responsabilă și declarată a asistenților AI. |

**9. Conținuturi**

| 9.1 Curs | Nr. ore | Metode de predare | Observații |
|---|---|---|---|
| 1. Introducere: ce este învățarea profundă și instrumentele de lucru. | 2 | Prelegere interactivă cu demonstrații live; predare „de sus în jos” (întâi modelul funcțional, apoi mecanismele); matematica prin cod; discuții și sondaje; materiale și teste de autoevaluare pe Campus Virtual. | 2 ore pe săpt.; laborator în aceeași săpt. Săpt. 1 fără cod. Teste în săpt. 7 și 11. |
| 2. Fluxul unui proiect de învățare automată: date, model, evaluare, livrare. | 2 | | |
| 3. Fundamente matematice prin cod: tensori și operații, difuzare, gradient, regresie liniară și logistică. | 2 | | |
| 4. Rețele neuronale multistrat și backpropagation. | 2 | | |
| 5. Antrenarea rețelelor: optimizare, regularizare, diagnosticarea antrenării. | 2 | | |
| 6. Învățare profundă pe date tabulare; metrici și referințe clasice. | 2 | | |
| 7. Rețele convoluționale. | 2 | | |
| 8. Învățare prin transfer în viziunea artificială. | 2 | | |
| 9. Secvențe și serii de timp: rețele recurente. | 2 | | |
| 10. Text: reprezentări vectoriale, clasificare, mecanismul de atenție. | 2 | | |
| 11. Arhitectura transformer și adaptarea modelelor preantrenate. | 2 | | |
| 12. De la caietul de lucru la produs: reproductibilitate, export, servire. | 2 | | |
| 13. Învățare profundă responsabilă: bias, protecția datelor, explicabilitate; redactarea raportului experimental. | 2 | | |
| 14. Sinteză și tendințe: modele generative, modele de fundație; pregătirea examenului. | 2 | | |

Bibliografie


1. Zhang, A., Lipton, Z. C., Li, M., Smola, A. J., Dive into Deep Learning, Cambridge University Press, 2023; ediția online gratuită, https://d2l.ai.
2. Antiga, L., Stevens, E., Huang, H., Viehmann, T., Deep Learning with PyTorch, ed. a 2-a, Manning, 2026.
3. Prince, S. J. D., Understanding Deep Learning, MIT Press, 2023; disponibilă gratuit la https://udlbook.github.io/udlbook/.
4. Bishop, C. M., Bishop, H., Deep Learning: Foundations and Concepts, Springer, 2024; ediție online gratuită, https://www.bishopbook.com.
5. Chollet, F., Watson, M., Deep Learning with Python, ed. a 3-a, Manning, 2025.
6. Raschka, S., Liu, Y., Mirjalili, V., Machine Learning with PyTorch and Scikit-Learn, Packt, 2022.
7. Howard, J., Gugger, S., Deep Learning for Coders with fastai and PyTorch, O'Reilly, 2020; caietele gratuite și cursul Practical Deep Learning for Coders, https://course.fast.ai.
8. Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, MIT Press, 2016; disponibilă gratuit la https://www.deeplearningbook.org (referință teoretică).
9. MIT 6.S191 Introduction to Deep Learning, https://introtodeeplearning.com; Karpathy, A., Neural Networks: Zero to Hero (video și cod), https://karpathy.ai/zero-to-hero.html.
10. Suportul de curs al disciplinei, publicat pe Campus Virtual.

| 9.2 Seminar / laborator / proiect / practică | Nr. ore | Metode de predare | Observații |
|---|---|---|---|
| 1. Mediul de lucru; prezentarea de deschidere realizată cu instrumente AI (L1). | 2 | Lucru individual, în perechi și în echipe, în caiete de lucru cu schele de cod și teste automate; sarcini pe două niveluri; învățare prin proiect; evaluare între colegi; asistenți AI ca tutor, declarați; predare prin Campus Virtual. | 2 ore pe săpt., 2 subgrupe. Bază pe CPU; cloud pentru extensii și proiect. Nivelare Python (până în săpt. 4) și tutoriat. Echipe de 3, mixte. |
| 2. Primul model prin învățare prin transfer, cu cod dat; Python științific (L2). | 2 | | |
| 3. Tensori, gradient și diferențiere automată; regresie liniară și logistică (L3). | 2 | | |
| 4. Perceptron multistrat de la zero, apoi cu framework-ul; fișa echipei de proiect (P0) (L4). | 2 | | |
| 5. Bucla de antrenare completă; experimente cu hiperparametri (L5). | 2 | | |
| 6. Studiu de caz pe date tabulare: referință clasică și rețea neuronală (L6). | 2 | | |
| 7. Rețele convoluționale pe seturi de imagini standard; augmentarea datelor (L7). | 2 | | |
| 8. Adaptarea unui model preantrenat pe un set propriu de imagini; explicabilitate (L8). | 2 | | |
| 9. Prognoză pe serii de timp cu rețele recurente (L9). | 2 | | |
| 10. Clasificare de text în română și engleză; propunerea de proiect (P1) (L10). | 2 | | |
| 11. Adaptarea unui model de limbaj preantrenat pentru clasificare (L11). | 2 | | |
| 12. Împachetarea unui model ca aplicație demonstrativă; script de antrenare reproductibil (L12). | 2 | | |
| 13. Lucru la proiect; evaluare între colegi a rapoartelor intermediare (P2). | 2 | | |
| 14. Prezentările proiectelor de echipă (P3). | 2 | | |

Bibliografie

1. Îndrumarul de laborator al disciplinei (caiete de lucru, șabloane de cod și de raport, reguli de utilizare a asistenților AI), publicat pe Campus Virtual.
2. Documentația și tutorialele oficiale ale framework-ului de învățare profundă utilizat (PyTorch, https://pytorch.org/tutorials) și ale bibliotecilor de modele preantrenate (Hugging Face, https://huggingface.co/docs).
3. Lippe, P., UvA Deep Learning Tutorials (caiete de lucru), https://uvadlc-notebooks.readthedocs.io.
4. scikit-learn, User Guide, https://scikit-learn.org.

**10. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului**

Conținuturile disciplinei urmăresc fluxul de lucru cerut în posturile de machine learning engineer, data scientist și AI engineer: Python și framework-urile actuale de învățare profundă, pregătirea datelor, antrenarea și evaluarea onestă a modelelor, învățarea prin transfer cu modele preantrenate, reproductibilitatea și livrarea modelului ca aplicație. Aceste cerințe se regăsesc în profilurile ocupaționale ESCO „data scientist” (ISCO 2511.3) și „ICT intelligent systems designer” (ISCO 2511.10; COR 251107, proiectant de sisteme informatice inteligente TIC), iar rapoartele recente asupra pieței muncii arată o creștere a cererii de competențe de inteligență artificială mult mai rapidă decât media pieței, inclusiv pentru posturile de intrare. Portofoliul de cod versionat și demonstrația publică a modelului sunt livrabile pe care angajatorii le pot verifica direct.

Disciplina este singura disciplină obligatorie de învățare automată din programul de master; ea

pregătește disciplinele Procesarea limbajului natural, Tehnici avansate de data mining, Metodologia cercetării științifice și elaborarea lucrării de disertație. Structura ei (predare aplicativă, învățare prin transfer, responsabilitate) urmează cursurile de referință actuale din comunitatea academică (fast.ai, MIT 6.S191, Stanford CS231n, Dive into Deep Learning).

**11. Evaluare**

| **Tip activitate** | **11.1 Criterii de evaluare** | **11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)** | **11.3 Pondere din nota finală** |
|---|---|---|---|
| 11.4 Curs | Cunoașterea și înțelegerea conceptelor (RI1-RI3): mecanismele rețelelor neuronale, arhitecturile fundamentale și adecvarea lor la tipul de date, metodologia experimentală, cerințele etice și legale.<br>Capacitatea de a construi, antrena, evalua și interpreta individual un model pe un set de date nevăzut (RI4, RI6, RI7). | Evaluare continuă: două teste pe Campus Virtual (săpt. 7, materia săpt. 1-6; săpt. 11, materia săpt. 7-10), corectate automat (10%).<br>Evaluare sumativă, examen în sesiune: probă practică individuală pe calculator (2 ore, fără asistenți AI) pe un caiet de lucru parțial scris și un set de date nevăzut, verificată automat și citită de titular (20%); chestionar conceptual pe Campus Virtual, 30 min (10%). | 40% (10% teste, 20% probă practică, 10% chestionar conceptual) |
| 11.5 Seminar/Laborator /Proiect / practică | Portofoliul de laborator (RI4-RI6, RI9): prezentarea de deschidere (L1) și caietele L2-L12, cu nivelul de bază trecut de testele automate și extensii rezolvate; respectarea termenelor; declararea asistenților AI.<br>Proiectul de echipă (RI3, RI7-RI9): formularea problemei, metodologie corectă (împărțirea datelor, fără scurgeri, referință clasică), calitatea modelului și a experimentelor, analiza erorilor, reproductibilitatea și calitatea codului, raportul și prezentarea, contribuția individuală vizibilă în istoricul codului. | Evaluare continuă: portofoliul L1-L12, predat prin Campus Virtual, verificat automat și notat pe rubrică, cu evaluare asistată de AI validată de titular (30%).<br>Evaluare continuă: proiectul de echipă, patru etape notate pe rubrică: P0 fișa echipei (săpt. 4, 5%), P1 propunere (săpt. 10, 15%), P2 raport intermediar cu evaluare între colegi (săpt. 13, 25%), P3 livrare finală (săpt. 14, 55%); nota individuală poate diferi de a echipei cu până la 2 puncte (30%). | 60% (30% portofoliu de laborator, 30% proiect de echipă) |

11.6 Standard minim de performanță
Nota finală N = 0,30 × PL + 0,30 × PR + 0,10 × TM + 0,20 × EP + 0,10 × EC, unde PL = portofoliul de laborator, PR = proiectul de echipă (cu ajustarea individuală), TM = media testelor de pe Campus Virtual, EP = proba practică, EC = chestionarul conceptual, toate notate de la 1 la 10. Evaluarea continuă (PL, PR, TM) reprezintă 70% din nota finală, evaluarea sumativă (EP, EC) 30%.
Condiții minime pentru nota 5: prezentarea de deschidere susținută și toate caietele L2-L12 predate, cu nivelul de bază trecut de testele automate, și PL ≥ 5; proiectul livrat integral (P0-P3), cu contribuție individuală vizibilă în istoricul codului, și PR ≥ 5; ambele teste de pe Campus Virtual susținute; la proba practică, studentul realizează singur un pipeline complet care rulează și produce o evaluare corectă pe setul de test, cu EP ≥ 5; nota finală N ≥ 5.
Livrabilele întârziate pierd un punct pe săptămână din nota livrabilului. În sesiunea de restanțe se reface numai examenul (proba practică și chestionarul conceptual); portofoliul, proiectul și testele rămân valabile în anul universitar curent. Asistenții AI se declară în fiecare livrabil și sunt interziși la examen; nedeclararea sau incapacitatea de a explica propriul cod se tratează ca abatere de la integritatea academică, conform regulamentelor UTCN.

| **Data completării:** | **Titulari** | **grad didactic, titlu Prenume NUME** | **Semnătura** |
|---|---|---|---|
| 10.09.2026 | Curs | Conf. univ. dr. Cosmin Nicolae SABO | |
| | Aplicații | Conf. univ. dr. Cosmin Nicolae SABO | |
| | | - | |

| | |
|---|---|
| Avizul Directorului departamentului care coordonează disciplina, doar dacă acesta diferă de departamentul organizator al programului de studii | Nu este cazul (disciplina este coordonată de departamentul organizator al programului de studii). |

| | |
|---|---|
| Data avizării în Consiliul Departamentului: 14.09.2026<br><br>_____ | Director Departament DMI<br>Conf. univ. dr. Cosmin Nicolae SABO |
| Data aprobării în Consiliul Facultății: 17.09.2026<br><br>_____ | Decan,<br>Conf. univ. dr. Monica Liliana MARIAN |