Lucrul la un proiect nu este obligatoriu, promovarea nefiind condiționată de punctajul la proiect. Fără proiect nota maximă ce poate fi obținută la Inteligență Artificială este 8.

O echipa de lucru la proiect poate avea maxim 5 persoane. Nu se acordă bonus suplimentar sau specificații mai relaxate pentru echipe mai mici de cinci persoane. Punctajele obținute de membrii echipei pot fi diferite și vor fi negociate la predarea proiectului.

Predarea unui proiect se face la o dată și locație ce vor fi precizate ulterior. Punctajele finale la proiect și echipele participante la Learn&Earn vor fi comunicate până cel târziu în ziua examenului.

Cele două teme asociate cu fiecare proiect vor fi rezolvate de echipele care au ales acel proiect. Studenții care nu au ales un proiect pot rezolva oricare din temele de mai jos, respectiv pentru laboratorul 12 pot prezenta oricare din temele “Tema 1” și pentru laboratorul 14 oricare din temele “Tema 2”.

Pentru alegerea unui proiect folosiți formularul de aici. Termen limită pentru alegerea proiectului: **8 decembrie**.

Termene limită teme:
Tema 1: lab 12 (18-22 decembrie)
Tema 2: lab 14 (15-19 ianuarie)

Proiect 1: ChatBotBuster

- Aplicația permite identificarea răspunsului, ales dintr-o pereche de răspunsuri la o întrebare cunoscută, cel mai probabil să fi fost parţial sau total generat automat.
- Pentru antrenarea unor clasificatori puteți folosi datele disponibile aici.
- Dacă utilizați rețele neuronale, construiți instanțe de antrenament ca perechi întrebare/răspuns om (instanță negativă) și întrebare/răspuns generat (instanță pozitivă).
- Dacă utilizați reinforcement learning, folosiți o funcție de reward ce măsoară distanța semantică dintre două texte, de exemplu cel de aici: https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder
- Descrierea unei soluții similare o puteți citi aici.

Tema 1

Implementați o interfață suport pentru un sistem ce utilizează reinforcement learning pentru a învăța cel mai bun răspuns ce poate fi oferit la o întrebare primită. Puteți folosi ca exemple de întrebări și răspunsuri datele de aici. Interfața afișează pe rând o întrebare și o pereche de răspunsuri, utilizatorul selectează unul din cele două răspunsuri și selecția contribuie la un scor asociat cu acele răspunsuri, scor ce poate fi folosit ulterior ca reward pentru antrenarea unui sistem de generare automată a acelor răspunsuri. Descrierea unei soluții similare o puteți citi aici.
(0.2p) Citirea și preluarea datelor din fișier.
(0.4p) Interfața ce afișează întrebarea și răspunsurile.
(0.4p) Selectarea răspunsului preferat și actualizarea scorurilor asociate cu acele răspunsuri.

Tema 2

Implementați o aplicație care generează versiuni alternative ale unui text primit la intrare înlocuind cuvinte din text cu sinonime sau hipernime extrase din Wordnet. Pentru limba română folosiți https://pypi.org/project/rowordnet/ , pentru limba engleză folosiți https://wordnet.princeton.edu/related-projects . Informații suplimentare găsiți aici.
(0.2p) Citirea textului original de la utilizator sau încărcarea lui dintr-un fișier.
(0.4p) Căutarea unui cuvânt în Wordnet și extragerea sinonimelor și hipernimelor sale.
(0.4p) Afișarea unei versiuni modificate a textului inițial, cu posibilitatea de a genera și alte versiuni la solicitarea utilizatorului.

Proiect 2: ChessExplained

- Aplicația permite utilizatorului să poarte un dialog în limbaj natural cu un expert AI în șah.

- Expertul poate rezolva și explica soluții (secvențe optime de mutări) pentru diverse configurații introduse de utilizator. Pentru identificarea secvențelor optime de mutări puteți folosi o bibliotecă existentă de genul Stockfish.
- Expertul poate explica regulile jocului de șah și notația algebrică pentru șah.

Tema 1

Adaptarea/antrenarea unui sistem de dialog pentru a deveni capabil să explice regulile jocului de șah. Puteți folosi oricare din următoarele trei alternative:

1. https://platform.openai.com/finetune
2. https://realpython.com/build-a-chatbot-python-chatterbot/
3. https://github.com/microsoft/botframework-sdk

Limba dialogului poate fi română sau engleză și sursa informațiilor oferite ca răspuns poate fi wikipedia (https://en.wikipedia.org/wiki/Rules_of_chess sau https://ro.wikipedia.org/wiki/Regulile_jocului_de_șah). Sistemul trebuie să fie capabil să răspundă corect la întrebări de genul "Ce înseamnă promovarea pionului?" cu răspunsul oferit de wikipedia "Dacă un pion ajunge pe linia a opta, acesta este promovat în damă, turn, nebun sau cal de aceeași culoare, alegerea fiind la discreția jucătorului (de obicei este aleasă dama).".

(0.4p) Pregătirea datelor pentru antrenarea sistemului conform specificațiilor.
(0.6p) Antrenarea folosind acele date și demonstrarea noilor capabilități.

Tema 2

Implementarea unui sistem capabil să descopere secvențe de cinci mutări ale unui jucător urmate frecvent de încheierea partidei. Folosiți exemplele de partide în notație algebrică normală de aici: https://drive.google.com/file/d/1diXVRm1AELi5888N-nivzqNN65yZQgyu/ . Partidele se termină cu semnul #. Pentru citirea partidelor puteți folosi biblioteci existente de genul: https://python-chess.readthedocs.io/en/latest/ . La ieșire sistemul afișează toate secvențele de cinci mutări ale unuia din jucători care apar la sfârșitul partidei de cel puțin 3 ori în setul de date oferit.
(0.3p) Citirea datelor din fișier utilizând o bibliotecă specifică.
(0.7p) Descoperirea și afișarea secvențelor cu frecvența menționată.

Proiect 3: SensorML

Datasetul SensorML conține date referitoare la evoluția în timp a unor parametri de mediu (temperatură, luminozitate, umiditate, presiune, etc.) pentru o cultură de tomate. Pornind de la acest dataset, realizați o aplicație care să:

- Prezică evoluția în timp a parametrilor de mediu prin antrenarea mai multor modele specifice seriilor de timp. Se vor utiliza (minim):
  - 1 model statistic (Prophet, (S)ARIMA, etc.)
  - 1 model bazat pe o rețea neuronală recurentă (LSTM, GRU, etc.)
  - 1 model de tip Seq2Seq

  Pentru implementarea modelelor puteți utiliza librării dedicate (de exemplu: PyTorch, TensorFlow, tsai, statsmodels, etc.).

- Ofere o interfață web care să cuprindă graficele corespunzătoare predicțiilor realizate și informații referitoare la boli posibile ale plantelor, simptomele asociate acestora și condițiile optime de dezvoltare a bolilor. În plus, folosind predicțiile pentru valorile parametrilor și condițiile optime de dezvoltare a bolilor, aplicația va prezice riscul de boală.

Legături utile

- Folderul corespunzător proiectului

Tema 1

Pornind de la datasetul SensorML_small, ce conține date referitoare la evoluția în timp a unor parametri de mediu (temperatură, luminozitate, umiditate, presiune, etc.) pentru o cultură de tomate, antrenați modelul Prophet pentru a prezice evoluția parametrilor indicați în dataset.

Cerințe:
(0.3) 1. Realizați analiza univariată a datelor pentru fiecare parametru. Interpretați graficele rezultate
- (0.1) Heatmap-uri cu valorile medii și mediane pe zi
- (0.2) Boxploturi pentru identificarea outlierelor și a distribuției valorilor

(0.2) 2. Determinați matricea de corelație și analizați care variabile sunt corelate direct și invers.

(0.2) 3. Antrenați modelul Prophet pentru a realiza predicții univariate pentru fiecare parametru. Având datele pentru 7 zile (168 de ore), modelul face predicții pentru următoarele 2 zile (48 de ore).
(0.1) 4. Realizați grafice care indică valorile predicțiilor realizate de model în paralel cu valorile actuale.
(0.2) 5. Afișati eroarea la antrenare utilizând mecanismul de cross validare pentru serii de timp.

Tema 2

Pornind de la informațiile de aici despre principalele boli și simptome de boli ale tomatelor, construiți o ontologie care să surprindă corect și complet aceste informații.
Cerințe:
(0.5) 1. Construirea ontologiei într-un editor de ontologii (Protege)
(0.3) 2. Popularea ontologiei - cu minim 10 instanțe (manuală sau semi-automată)
(0.2) 3. Validarea ontologiei folosind un reasoner

Legături utile
- Resurse

Proiect 4: FeedbackHHC

- Analiza opiniilor pacienților cu privire la tratamentul acordat de către agenția de furnizare a serviciilor de asistență medicală la domiciliu.
- Setul de date este disponibil la adresa https://data.cms.gov/provider-data/dataset/6jpm-sxkc . Setul de date conține informații referitoare la agenția care oferă servicii de asistență la domiciliu iar variabila target este calitatea îngrijirii pacientului. O analiză similară se găsește la adresa https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9407698/ .
- Utilizarea altor variabile pentru predicție: spre exemplu, utilizarea notelor clinice pentru a prezice spitalizarea pacientului https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7606545/ .

Tema 1: Selecția celor mai influente atribute
(0.1) 1. Preprocesarea datelor
(0.3) 2. Analiza exploratorie a setului de date: valorile medii și mediane, vizualizarea datelor sub forma unor histograme/ barplot-uri, etc.
(0.6) 3. Selecția atributelor (utilizați spre exemplu PCA, Correlation-based Feature Selection, etc.)

Tema 2: predicția feedbackului pacienților privind calitatea serviciilor oferite de agenție
Problema considerată este o problemă de clasificare multi-clasă.
(0.6) Antrenați cel puțin 2 algoritmi de clasificare (spre exemplu, rețele neuronale, Random forest https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html , etc.)
(0.3) Testați algoritmii antrenați și afișați metricile de performanță

(0.1) Realizați o analiză comparativă a algoritmilor testați (spre exemplu utilizând ROC (receiver operating characteristic curve)

Proiect 5: ElderlyChatbot

- Identificarea declinului cognitiv la persoanele în vârstă
- Implementarea unui chatbot pentru divertisment și monitorizare terapeutică a persoanelor în vârstă
  Chatbotul va fi capabil să:
  - să furnizeze știri din diferite domenii de interes ale utilizatorului
  - să genereze întrebări pentru a evalua dacă utilizatorul a înțeles știrea, memoria pe termen scurt a acestuia, dacă utilizatorul a fost concentrat în timpul conversației, etc.
  - să fie empatic cu utilizatorul
  Comunicarea cu chatbotul poate fi în scris sau orală.
- Descrierea unei soluții similare: https://link.springer.com/article/10.1007/s12652-022-03849-2

Tema 1: Utilizarea tehnicilor de NLP și învățare automată pentru generarea de text.
(0.2) Extragerea știrilor recente din anumite domenii, în funcție de preferințele utilizatorului.
Poate fi utilizată o librărie (vezi https://pypi.org/project/news-fetch/ sau aici) sau un set de date (MIND, ag news).
(0.8) Generarea de întrebări și răspunsuri referitoare la știrile selectate.
Pot fi folosite tehnici de procesarea limbajului natural (spre ex. pentru extragerea de entități: nume, organizații, locații, etc.) sau rețele neuronale (aici sau aici sau puteți antrena un Transformer pe un set de date cu știri).

Tema 2: Implementarea unei interfețe suport pentru un sistem de detecție a declinului cognitiv. Folosiți exemple de întrebări generate în Tema 1. Interfața afișează pe rând întrebările și calculează similaritatea între răspunsul utilizatorului și răspunsul target. Utilizatorul poate comunica răspunsurile în scris sau oral.
Implementarea unor tehnici pentru detectarea declinului cognitiv din răspunsurile furnizate de utilizator.
(0.3) Interfața ce afișează întrebările
(0.3) Preluarea răspunsului utilizatorului (în scris sau orală)
(0.4) Calculul similarității între răspunsul utilizatorului și răspunsul target
Exemplu: https://huggingface.co/tasks/sentence-similarity