



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Villamosmérnöki és Informatikai Kar

# MÉLYTANULÁS

## NAGY HÁZI FELADAT

Model Ensemble for Medical Image Segmentation

Hallgatók: Berkovics Fanni, Jankó Júlia, Laczkó Anna

Konzulens: Kalapos András

### Tartalomjegyzék

<b>1 Bevezetés</b>	<b>2</b>
1.1 Adathalmaz	2
1.2 LLM használat	2
1.3 Feladatmegosztás	2
<b>2 Kód dokumentáció</b>	<b>3</b>
2.1 Rendszerkövetelmények	3
2.2 Data Preparation.ipynb	3
2.3 Training and Evaluation.ipynb	4
2.4 User Interface.ipynb	6
2.5 Docker	3
<b>3 Irodalomjegyzék</b>	<b>7</b>

2023/2024. 1. félév

# 1 Bevezetés

A kitűzött feladat célja egy modellegyüttes létrehozása, amely képi adatok alapján klasszifikációt valósít meg. A modellegyüttesek [1] használata bevett módszernek számít mind a statisztikába, mind a gépi tanulásban; több tanulási algoritmust használnak, hogy jobb prediktív teljesítményt érjenek el, mint amelyet az alkotó tanulási algoritmusok önmagukban elérnének.

## 1.1 Adathalmaz

A Sunnybrook Cardiac Data (SCD) [2] egy 45 MRI-felvételt tartalmazó, nyilvánosan elérhető adatkészlet. A felvételek négy patológiás csoportból (egészséges, hipertrofia, szívelégtelenség infarktussal és szívelégtelenség infarktus nélkül) származnak.

## 1.2 LLM használat

A feladat elvégzése során előfordult, hogy igénybe vettük a ChatGPT ingyenesen elérhető változatát, ez jellemzően a debuggolást segítette vagy a kódminőség javítását szolgálta. A dokumentáció elkészítésénél nem használtuk.

## 1.3 Feladatmegosztás

A mérföldköveket sprinteknek tekintve osztottuk ki egymás között a feladatokat, különös figyelmet fordítva a hallgatók aktuális elérhetőségére, preferenciáira, ugyanakkor az egyenlő terhelést szem előtt tartva.

Berkovics Fanni az első sprintben az adatgyűjtéssel és -elemzéssel, míg a második és a harmadik sprintben a kiértékeléssel és a vizualizációval foglalkozott.

Jankó Júlia az első sprintben a konténerizációval, míg a második és harmadik sprintben a baseline és a végső modellek kiválasztásával és betanításával foglalkozott.

Laczkó Anna az első sprintben az adatok előkészítésével és a GitHub létrehozásával, míg a második és harmadik sprintben az adatok effektív betöltésével és a felhasználói interfész kialakításával foglalkozott.

## **2 Kód dokumentáció**

### **2.1 Rendszerkövetelmények**

A notebookok a legtöbb lokális környezetben lefuttathatóak, a Docker használatához a README.md fájlban leírtakat kell követni. A használata erősen ajánlott, mivel ebben már telepítésre kerültek a szükséges Python csomagok a megfelelő verziókkal. Az egyetlen fontos kritérium, hogy az adott gépnek legalább 8 GB RAM-mal kell rendelkeznie.

### **2.2 Docker**

A konténerizációval kapcsolatos feladatok elvégzése eleinte nehézkes volt, mivel egyikünknek sincs sok tapasztalata vele. A Dockerfile tartalma többek között a ChatGPT segítségével készült el. Egy docker image-et készítettünk a requirements.txt és a Dockerfile felhasználásával, majd ezt feltöltöttük egyikünk docker hub-jára, hogy ne kelljen mindenkinek docker image-et építeni. Ennek használására az instrukciók a README.md fájlban vannak.

### **2.3 Data Preparation.ipynb**

Ez a notebook felelős az adatok előkészítésért és az effektív betöltésért. Az adatok effektív betöltése kiemelt szerepet kapott a projekt során, az adatok mennyiségének köszönhetően. A megoldás az lett, hogy az adatbetöltés minden lépése függvényként lett lekódolva, és összesen egy cella végzi az egész adatelőkészítés műveletét.

Ezen felül az adatok 12 db külön batch-ben kerülnek feldolgozásra, melyeket a script külön-külön is elment, hogy ne terheljük meg túlságosan a RAM-ot. Később a tanítás során ez a 12 fájl kerül betöltésre egyesével a feldolgozáshoz.

A .csv formátumban tárolt adatok feldolgozása során az életkort tartalmazó „age” jellemzőt csoportokra bontottuk, majd dummy változót hoztunk létre belőle, míg a nemet jelölő „gender” változóból flaget készítettünk.

## 2.4 Training and Evaluation.ipynb

Képi (szekvenciális) és táblázatos (numerikus) adataink is voltak, eleinte olyan modellt kerestünk, amely tud ezen két adaton egyszerre tanulni, azonban nem sikerült ilyen modellt találnunk. A baseline modell így két RandomForestClassifier-ből állt; egyik dolgozta fel a képi adatokat, míg a másik a táblázatosakat.

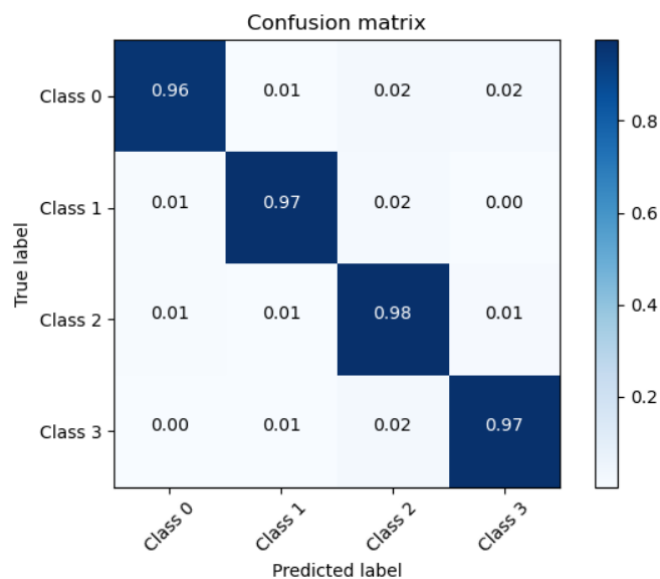
Az ensemble megoldás kivitelezéséhez a VotingClassifiert választottuk erre, soft szavazási móddal, nehogy a gyengébb, táblázatos adatokon tanult modellek elvigyék a döntést rossz irányba. A VotingClassifierbe egész pipeline-okat is be lehet adni egy-egy modellnél, így elég volt az egész dataframe struktúráját beadni tanító adatként, és az egyes modellek kiszedték az image vagy táblázatos adatokat belőle maguknak.

Inkrementálisan, batchenként tanítottuk be a modelleket, a tanító adat mérete miatt. Végző modellegyüttesünkben meghagytuk az eredendő RandomForestClassifier-eket. Egy GradientBoostingClassifiert alkalmaztunk a táblázatos adatokra, mivel a képi adatokból egy batch-re több, mint 40 perc kellett volna neki. A képi adatokra még egy egyszerűbb neurális hálót tanítottunk be, mivel ebbe képi transzformációs rétegeket beletéve teljesen elszállt volna a memóriahiánytól.

A modellek betanítására az adathalmaz 75%-át használtuk. A szavazásos modellegyüttes betanítására és a tesztelésre 16% és 9% arányban adtunk adatokat.

Manuálisan optimalizáltuk a hiperparamétereket, mivel az idő nagy része az adatok memóriába való beleféréssel telt el. Amennyiben nem adtunk megfelelő mennyiségű fát a RandomForestRegressor-nak, illetve a GradientBoostingClassifier-nek, nagyon rossz teljesítményük volt. Illetve az is szükséges volt az inkrementális tanításnál, hogy még több fát adjunk batch-enként, hogy ne felejtse el a régen tanultakat, és az újakat is be tudja fogadni.

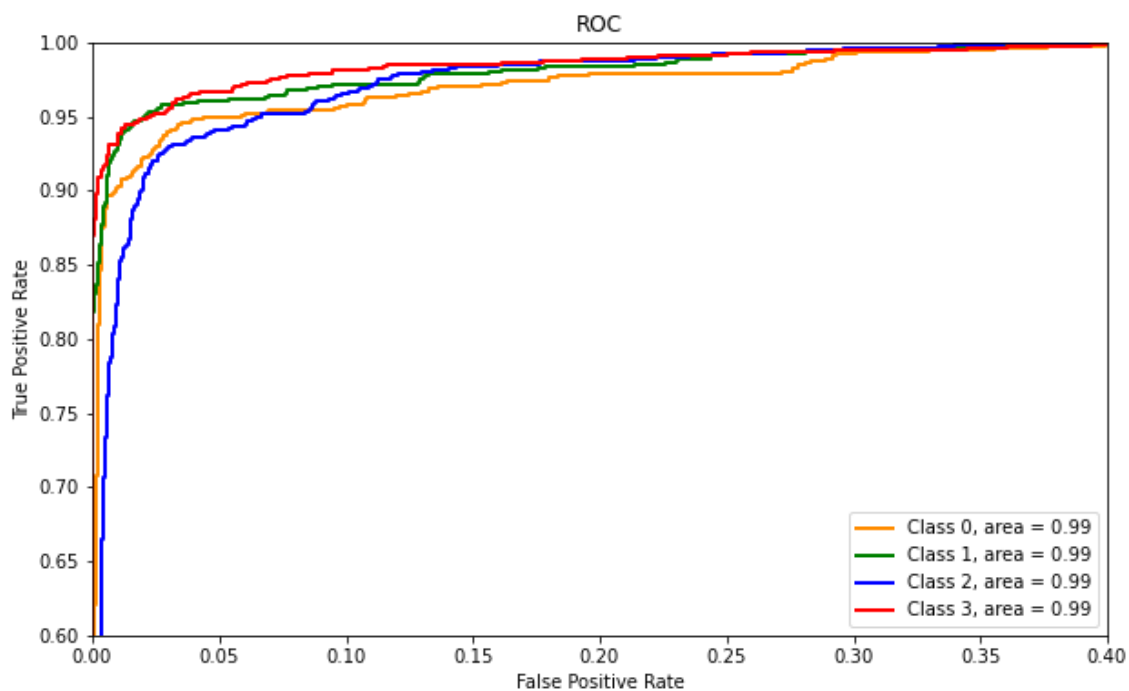
A modellegyüttes teljesítményének mérésére olyan metrikákat kerestünk, amelyek alkalmasak többosztályos klasszifikációs modellek teljesítményének értékelésére. Ide sorolható az összes olyan mutató, amelyet a konfúziós mátrix értékeiből nyerhetünk ki. A modellegyüttes predikcióihoz tartozó konfúziós mátrixot az 1. ábra szemlélteti.



**1. ábra: Konfúziós mátrix**

A mátrix értékeiből számos mutatót származtathatunk; accuracy (0.9679), precision (0.9680), recall (0.9679), F1 Score (0.9679), Cohen-féle kappa (0.9570), Matthews-korrelációs együttható (0.9570).

A klasszifikációs modellek értékelésénél a ROC-görbe vizsgálata is elengedhetetlen, alakulása a különféle osztályok esetében a 2. ábrán látható. Mivel a négy osztály esetében nagyon hasonló görbe rajzolható ki, így jelen ábra csak azt a tartományt mutatja, ahol az eltérések a legjobban megfigyelhetők.

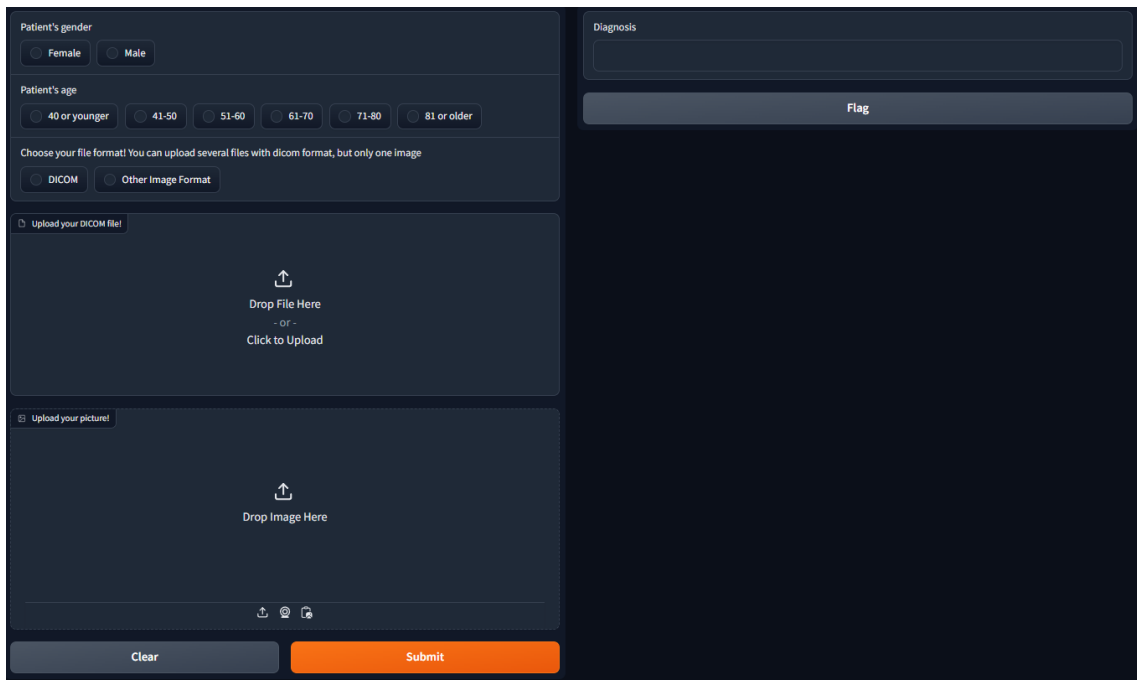


**2. ábra: ROC-görbe**

Összességében tehát elmondható, hogy a modellegyüttes jó eredményeket ér el, teljesítménye megfelelő, legyőzi a baseline modellt.

## 2.5 User Interface.ipynb

Felhasználói interfész kialakításához a Gradio applikációt használtuk. Ezen notebook lefuttatásának végén egy új ablakban automatikusan megnyílik a 3. ábrával megegyező felhasználói interfész.

The screenshot shows a web-based user interface for a medical image analysis application. It features a dark-themed layout with several input sections on the left and a results area on the right. The input sections include: 'Patient's gender' with radio buttons for 'Female' and 'Male'; 'Patient's age' with radio buttons for '40 or younger', '41-50', '51-60', '61-70', '71-80', and '81 or older'; a file format selection with 'DICOM' and 'Other Image Format' options; a large 'Drop File Here' area for uploading DICOM files; and another 'Drop Image Here' area for uploading pictures. At the bottom of the input section are 'Clear' and 'Submit' buttons. The right side of the interface contains a 'Diagnosis' text box and a 'Flag' button.

3. ábra: Felhasználói interfész

Először meg tudjuk adni a páciens adatait, majd kétféle képfeltöltési módból választhatunk; vagy egy darab képet tölthetünk fel, amit aztán kiértékel a program, vagy egy vagy akár több DICOM fájlt. Több fájl esetén a modell minden egyes képre prediktál egy címkét, majd a móduzt nevezi meg diagnózisnak. Emellett ilyenkor megjelenít egy százalékot is, melyet az alábbi képlet szerint számol:

$$\frac{\text{count(módusz)}}{\text{len(képek)}}$$

Ez a képlet könnyen változtatható igény esetén, és akár a megfelelő modellek `predict_proba` értékével is helyettesíthető.

### **3 Irodalomjegyzék**

[1] <https://scikit-learn.org/stable/modules/ensemble.html>

[2] <https://www.cardiacatlas.org/sunnybrook-cardiac-data/>