Clasificarea semnalelor EKG

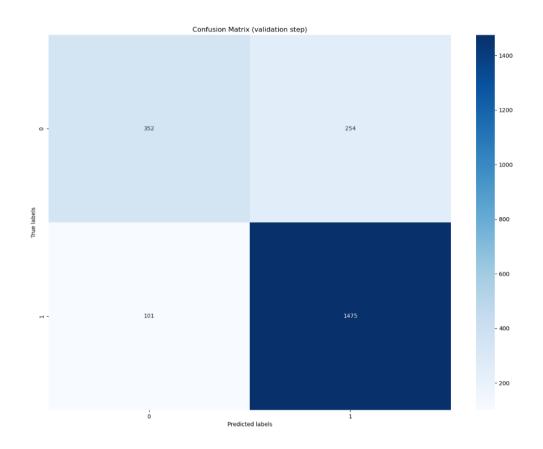
Concret dorim clasificarea semnalelor EKG primite intr-un fisier de tip csv in semnale sanatoase/nesanatoare. Astfel am folosit un set de date care are pe ultima coloana valori de 0/1 corspunzatoare pentru label-urile sanatos/bolnav. Setul de date folosit impartise datele in 2 fisiere separate intre sanatos/nesanatos, dar am considerat mai eficient sa unesc cele 2 fisiere in unul singur si sa folosesc label-ul de pe ultima coloana. Setul de date final are aproximativ 14500 de semnale care au fost impartite in date de antrenare, testare si validare astfel: 70% antrenare, 15% validare si 15% testare.

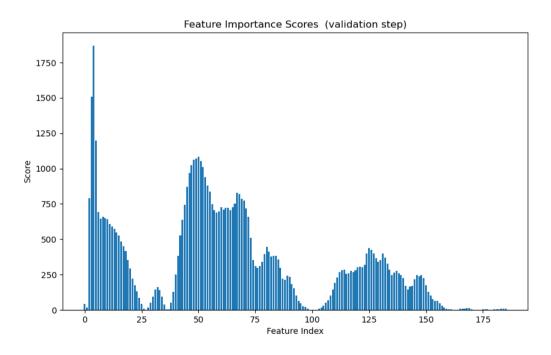
Am folosit algoritmul LogisticRegression din libraria sklearn folosind metoda "newton-cholesky" pentru solver. A fost ales acest algoritm deoarece este specializat pe clasificare binara (intre 2 clase ie: sanatos/bolnav).

In faza de validare am afisat valorile pentru Acuratete, Precizie, Sensibilitate (Recall) si Scorul-F1, insa am afisat si Matricea de confuzie, plot cu Feature importance scores, Curba ROC si AUC, Curba de invatare, curba Precizie-Recall si rezultatele predictiei comparate cu rezultatele reale.

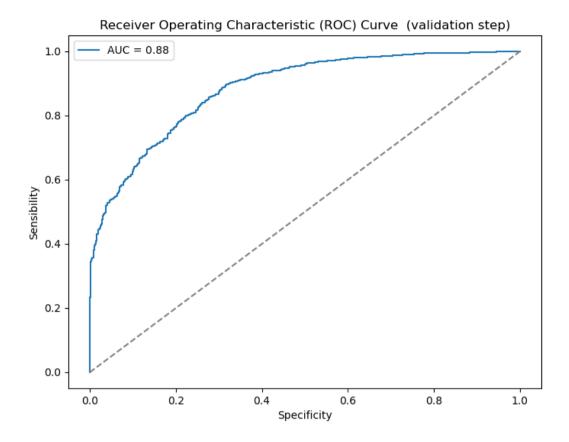
In faza de testare am afisat acuratetea, Matricea de confuzie si rezultatele predictiei comparate cu rezultatele reale.

Mai sus putem observa rezultatele modelului care are o acuratete de aproximativ 82%, ceea ce este bine tinand cont ca algoritmul ruleaza in 7 secunde pe un procesor foarte putin performant. Recall-ul are o valoare mica, ceea ce nu este de dorit in aceasta aplicatie deoarece consider mai important ca algoritmul sa arate cazuri sanatoase ca bolnave, decat invers deoarece aceste erori pot duce la pierderi de vieti.

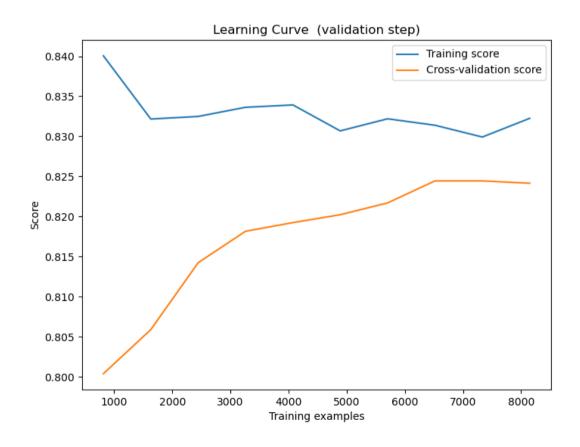


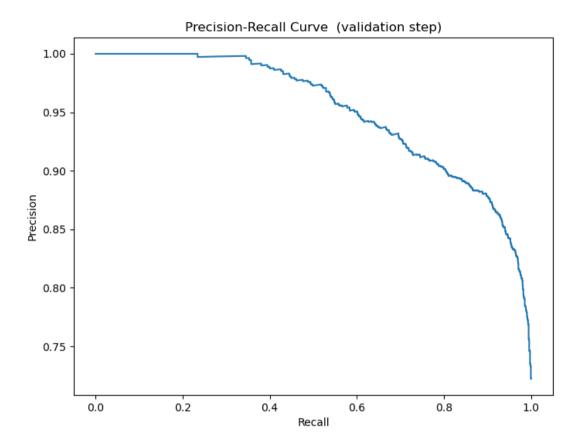


Graficul de mai sus arata cat de mult a contat in antrenare fiecare coloana a fisierului, fiind de asteptat importanta redusa a coloanelor de la 160 la final deoarece multe semnale din fisier nu acopera portiunea respectiva, fiind completate cu zerouri pentru a fi uniforma dimensiunea. Inca nu am gasit o explicatie pentru importanta redusa a coloanelor 25, 35 si 100.

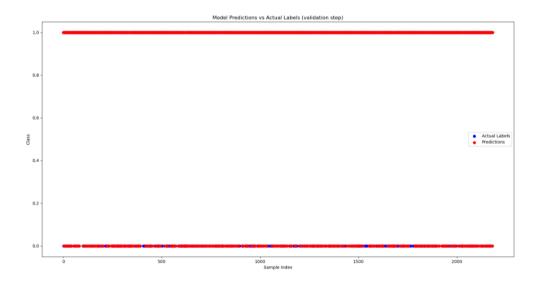


Deoarece AUC este aproape de 90% consider ca clasificatorul are performante bune.

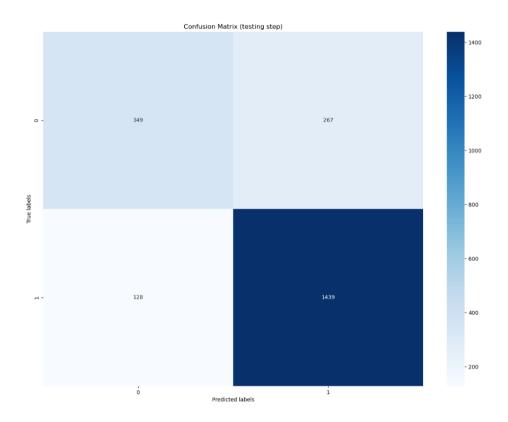


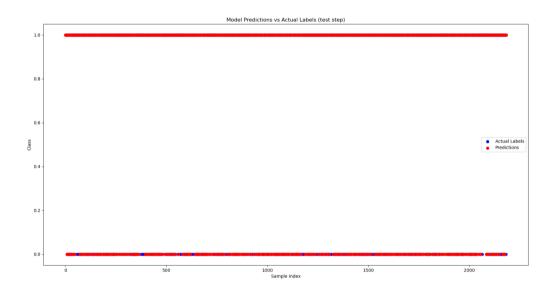


Aria de sub curba Precision-Recall este destul de mare, ceea ce confirma performantele clasificatorului.



Graficul de mai sus arata rezultatele grafice ale clasificarii, punctele albastre vizibile fiind clasificarile gresite ale clasificatorului.





Mai sus avem un exemplu de rezultat pe setul de testare din datele initiale.

Programul permite si optiunea introducerii de noi date si clasificarea acestora, in cazul exemplului din cod acestea avand lable-uri si se poate verifica corectitudinea datelor.

Dupa cum se poate observa acuratetea nu este foarte mare, caz ce reflecta probabil faptul ca setul de date nu este ideal, dar nu pot spune cu certitudine deoarece nu sunt specialist in domeniul cardiologiei. Avem insa avantajul ca acest algoritm are un timp de executie mic (~7 secunde) si ar putea usura munca medicilor care trebuie sa verifice totate semnalele manual, si sa ii ajute sa filtreze semnalele sanatoase.

Bibliografie:

Setul de date: https://www.kaggle.com/datasets/shayanfazeli/heartbeat

Librarii folosite: pandas, sklearn, matplotlib, numpy, seaborn