

## Clasificarea semnalelor EKG

- continuare -

În documentul temei 1 se pot găsi descrierea temei, preprocesarea datelor și modul de împărțire a acestora.

Pentru această temă am modificat codul temei 1 și am folosit algoritmi indicați în cerință și anume K-Nearest Neighbors și Naive Bayes. Ambii algoritmi au fost introduși într-un singur fișier python pentru a putea compara rezultatele celor 2 algoritmi.

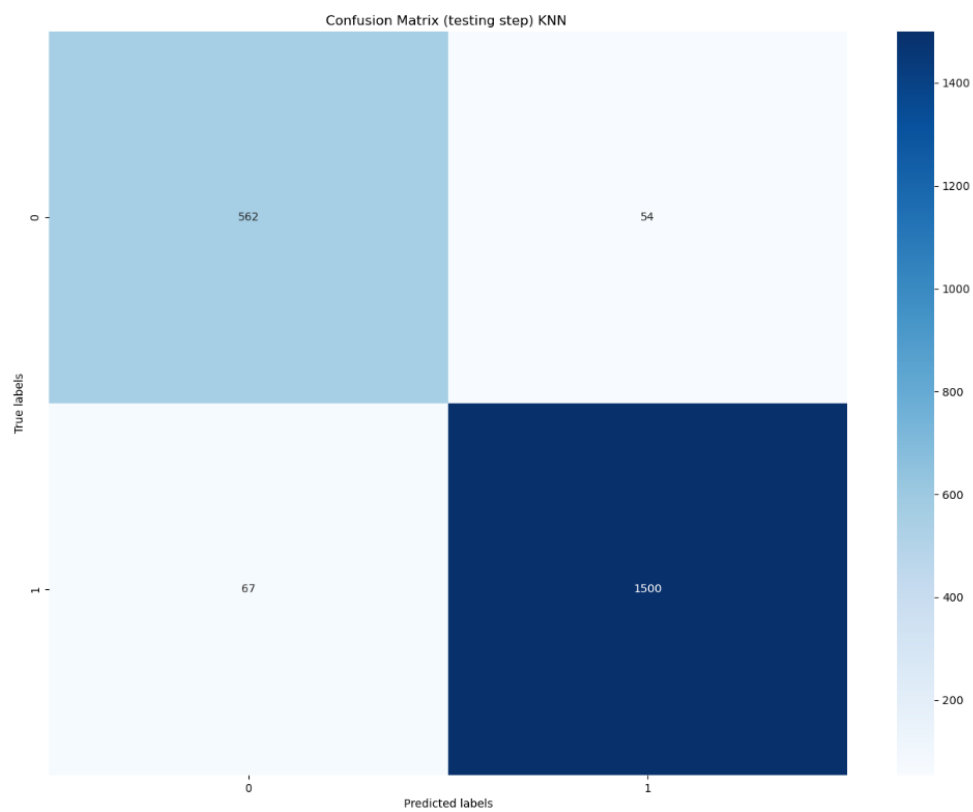
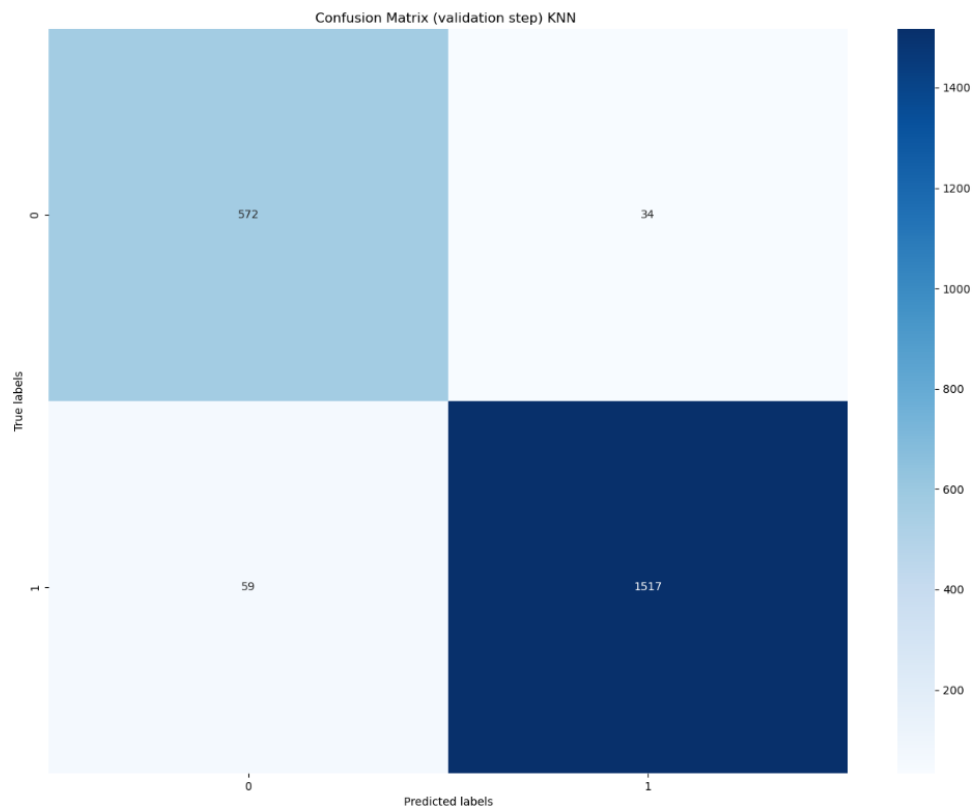
|               | KNN                | Naive Bayes        |
|---------------|--------------------|--------------------|
| Accuracy      | 0.9573785517873511 | 0.6315307057745188 |
| Precision     | 0.942288140875321  | 0.6557574111175252 |
| Recall        | 0.9532289206077968 | 0.6921102008678025 |
| F1 score      | 0.9475385712723485 | 0.6203778719514729 |
| Test Accuracy | 0.9445716903344022 | 0.6110856619331195 |

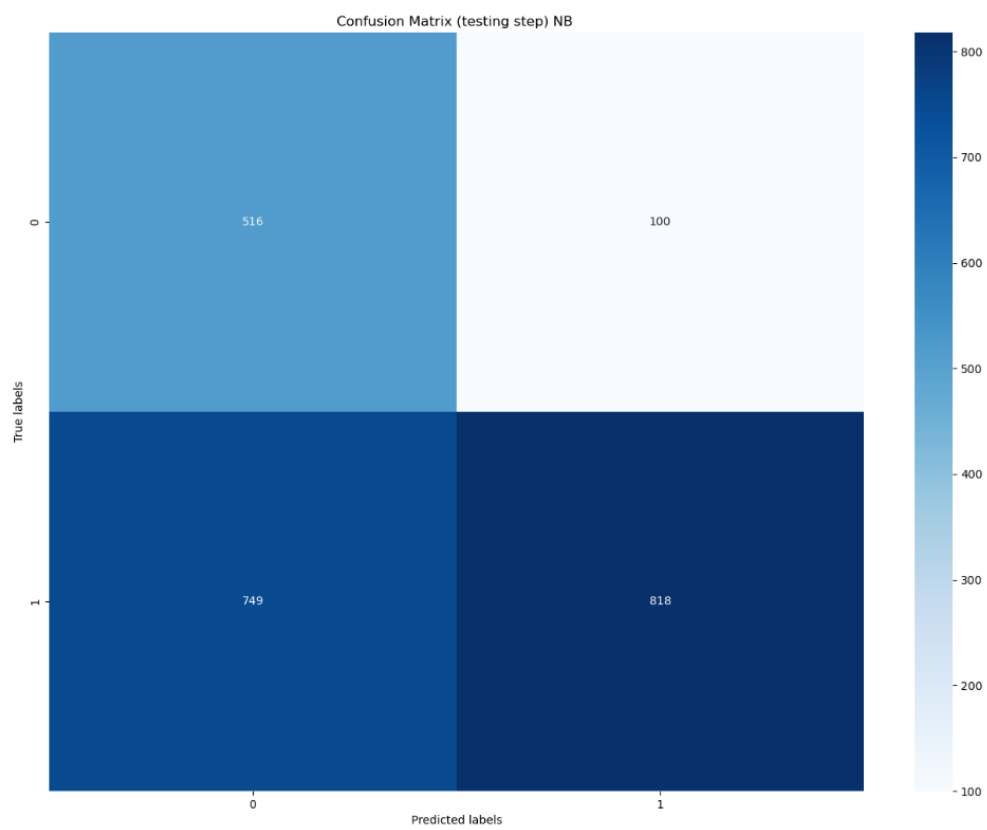
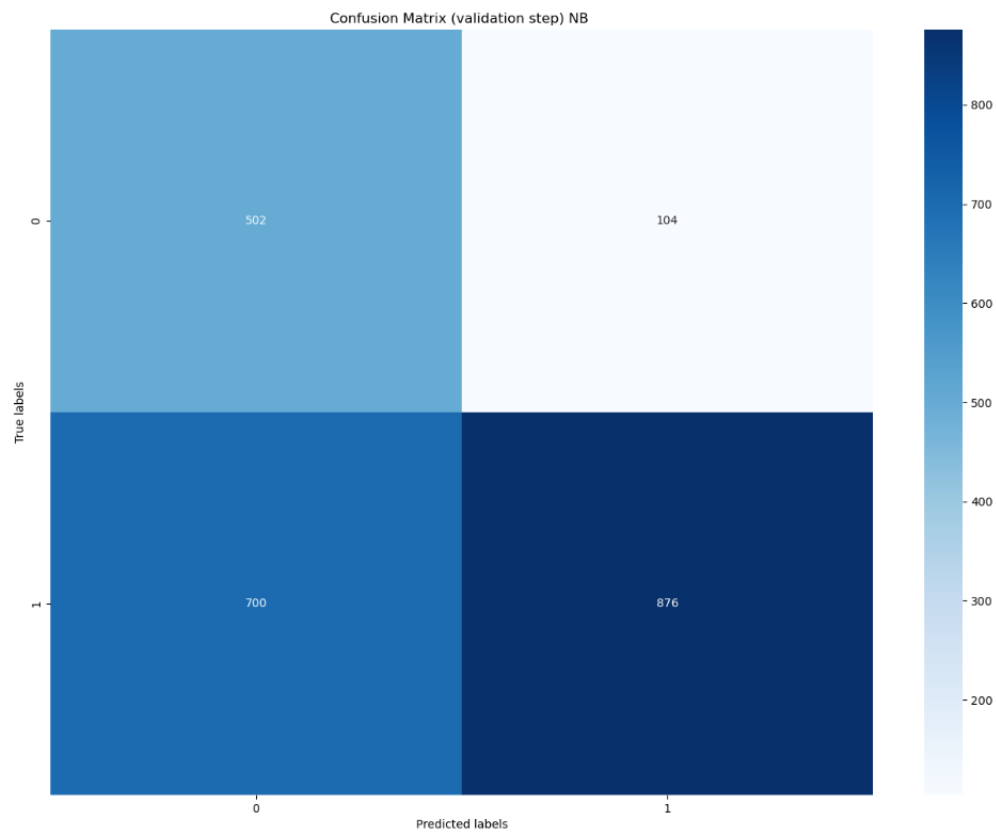
```
Validation stage
Accuracy: 0.8373052245646196
Precision: 0.8150681083758811
Recall: 0.7583858956961687
F1: 0.7786825409750586
```

```
Testing step
Accuracy: 0.8190563444800733
```

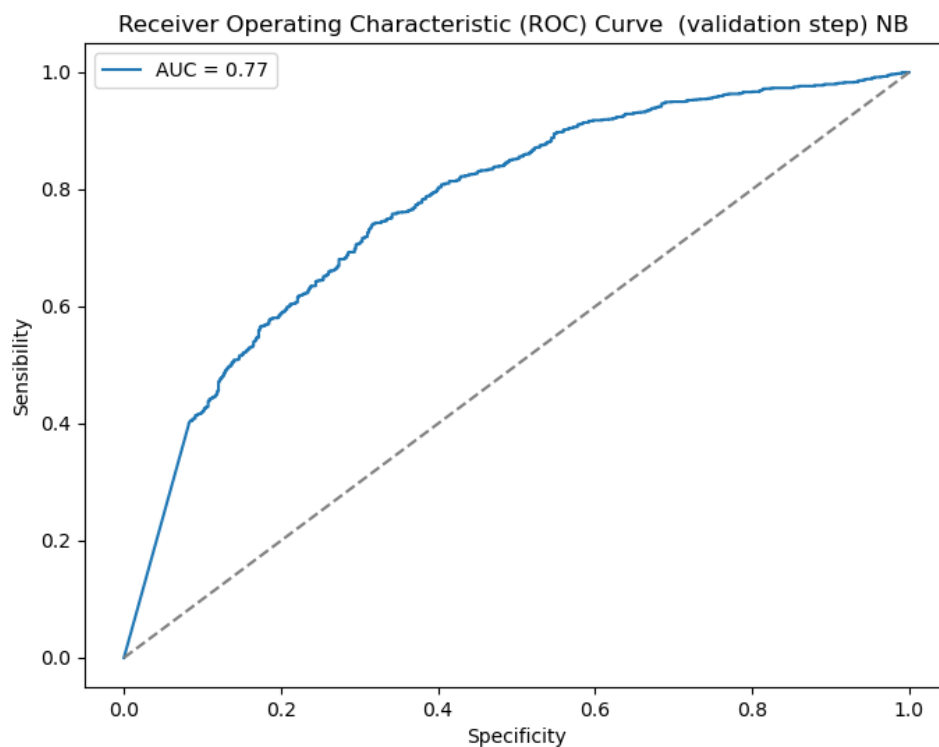
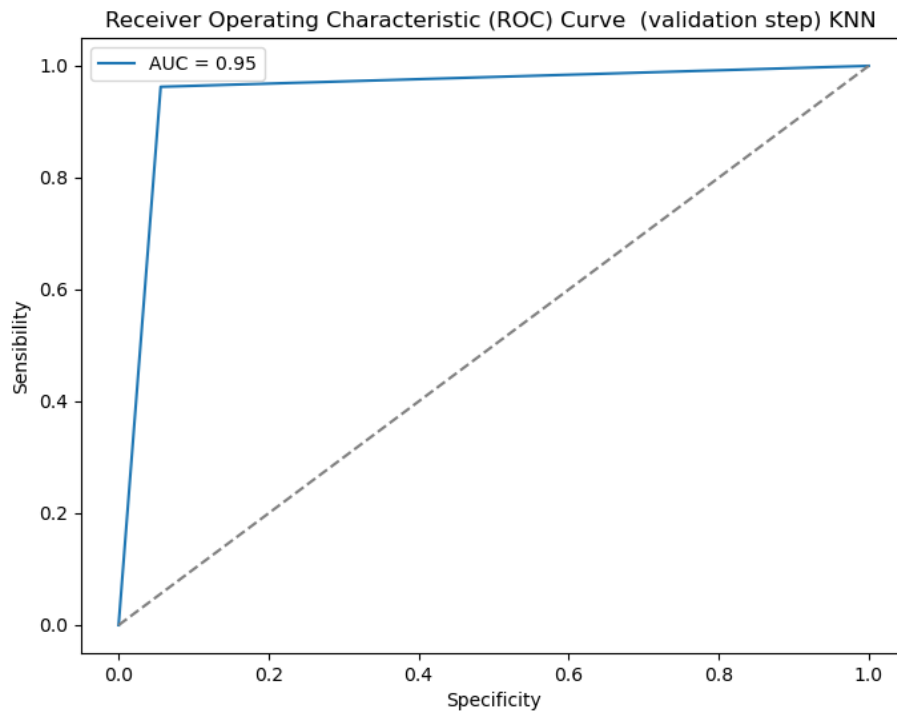
Din datele de mai sus se poate observa ca algoritmul KNN este cel mai eficient, întrecând cu mult chiar si algoritmul de regresie logistica folosit in tema 1.

Algoritmul KNN a fost configurat cu 1 n\_neighbors ceea ce a oferit cea mai buna performanță din toate variantele testate (n\_neighbors = 1:10). Precizia de aproape 96% este foarte buna pentru aceasta aplicatie, iar recall-ul care are o valoare de ~95% este important deoarece in aceasta aplicatie este de preferat sa apara cazuri sanatoase ca bolnave, si nu invers.



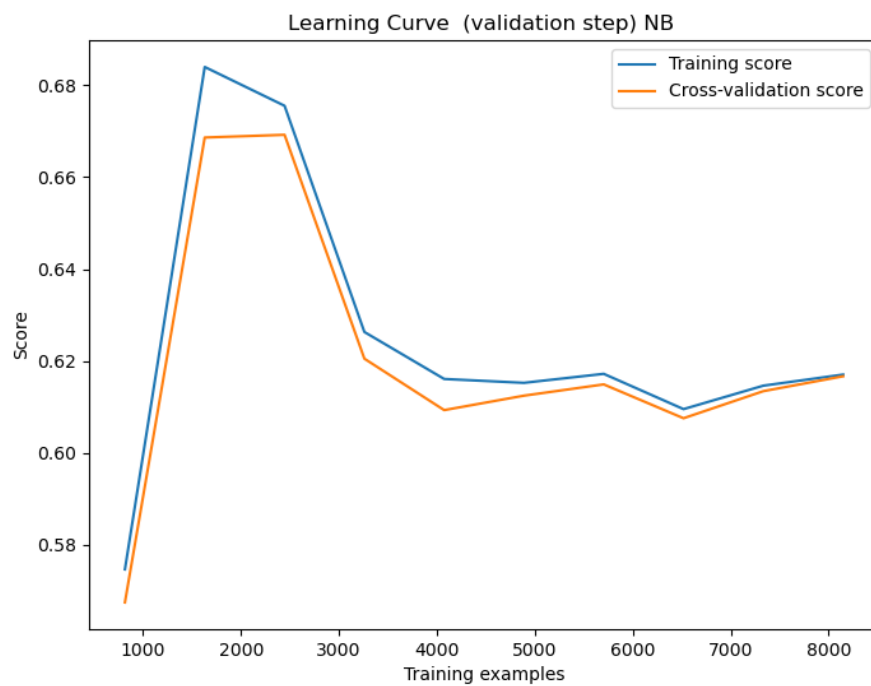
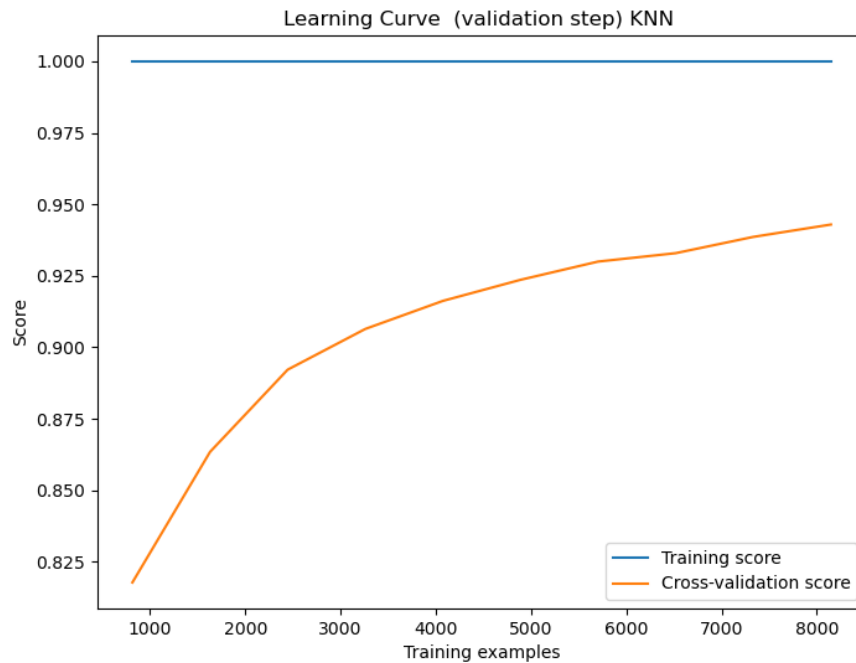


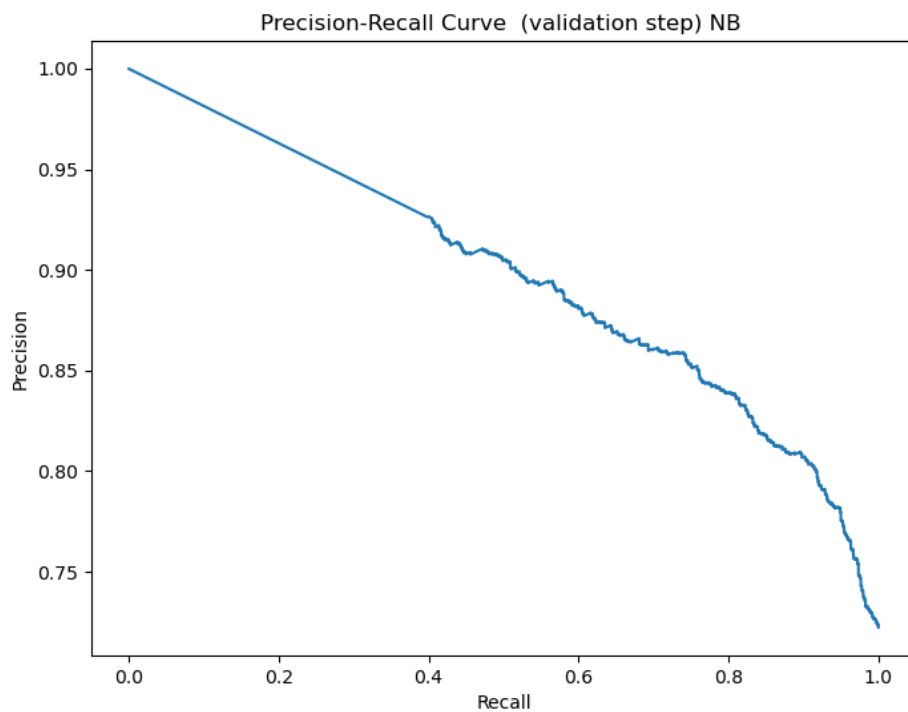
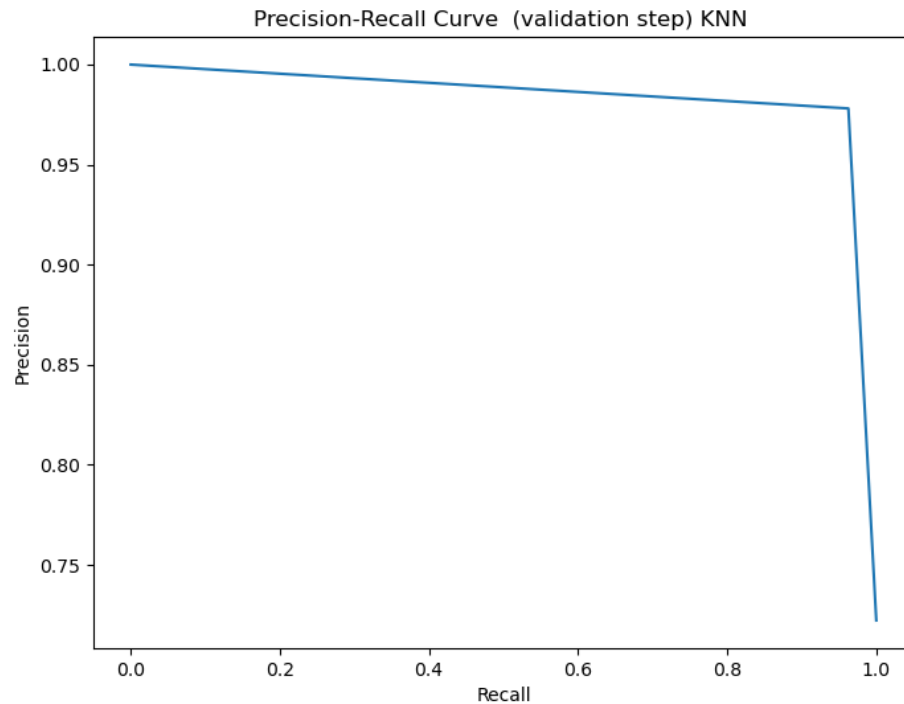
Mai sus putem vedea matricele de confuzie ale celor 2 algoritmi in fazele de validare si testare, cu rezultate similare intre cele 2 faze, dar cu diferente vaste intre ei.



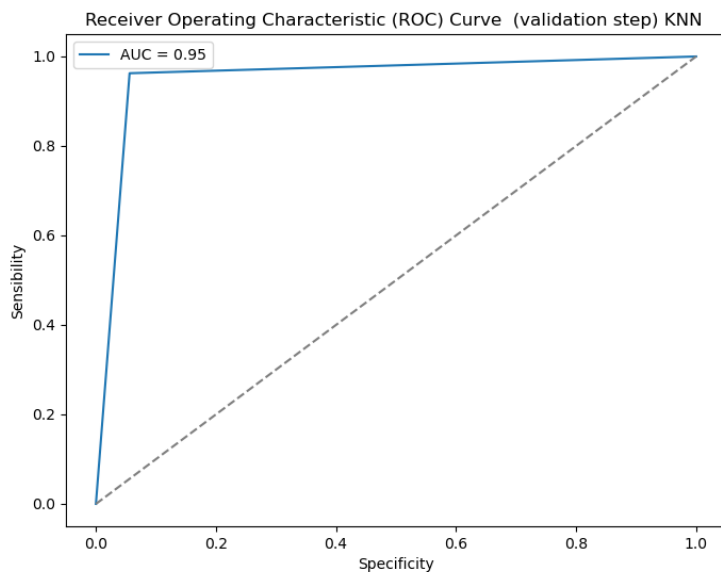
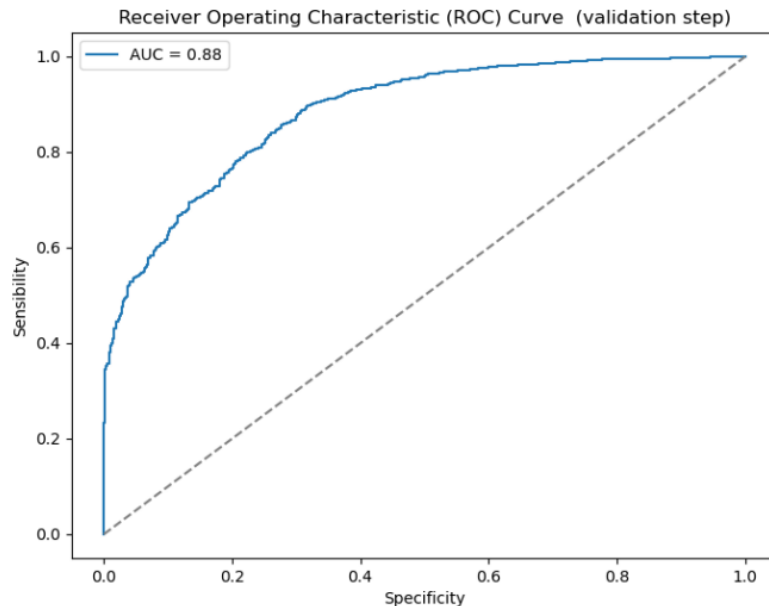
AUC pentru algoritmul KNN este 0.95, aproape de perfect, daca ne intereseaza AUC putem modifica numarul de `n_neighbors` intr-un numar mai mare de 2 dar mai mic ca 10 si

scorul va fi 0.97. AUC al algoritmului Naive Bayes este 0.77, ceea ce nu este foarte rau, inasa demonstreaza ca algoritmul NB este inferior algoritmului KNN in aceasta aplicatie.





Toate graficele de mai sus demonstreaza superioritatea algoritmului KNN in aplicatia noastra comparativ cu algoritmul Naive Bayes. KNN este mai eficient si decat algoritmul de regresie liniara folosit initial pentru rezolvarea acestei teme, fapt evidentiat foarte clar de curba ROC a celor 2 clasificatori.



Programul a pastrat functionalitatea de a testa cu date noi, oferind optiunea folosirii algoritmului KNN sau Naive Bayes.

In concluzie aceasta ajustare a modelului folosit a avut un impact mare asupra tuturor metricilor de calitate ale programului. Masurile de performanta au fost imbunatatite cu valori intre 14% și 20%. Astfel folosind algoritmul KNN toate masurile folosite in clasificare au valoarea de aproximativ 95%. Cu asemenea performante algoritmul ar putea fi luat in considerare pentru folosirea in domeniul medical daca i se atașează si un script care sa transforme datele primite de la aparatul EKG in fisiere cu extensia csv. Principala problema care ar putea aparea este faptul ca algoritmul a folosit un set de date produs de un anumit aparat, date care ar putea varia intre aparatele folosite, mai ales in cazul altor modele. In cazul unei aplicatii reale am avea nevoie de un set de date de la fiecare aparat pentru verificarea performatelor modelului, si adaptarea acestuia.

Bibliografie:

Set de date: original <https://www.kaggle.com/datasets/shayanfazeli/heartbeat>,  
insa a suferit mici modificari pentru a functiona corect

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