Documentul de Proiectare a Soluţiei Aplicaţiei Software

(Software Design Document)

Platforma pentru determinarea inferenței pentru

pacienții suspecți de COVID

ECHIPA: ÎNDRUMĂTOR:

Cosmin Daniela, 342C5 Vlăsceanu Georgiană

Cireașă Roxana, 342C5

Pribeagu Simona, 342C5

Florea Cosmin, 342C5

Bogozi Emilia, 341C5

Palade Cătălin, 342C5

Dragomir Adrian, 342C5

Contents

[Scopul documentului 3](#_Toc58319505)

[Conţinutul documentului 3](#_Toc58319506)

[Modelul datelor 3](#_Toc58319507)

[Structuri de date globale 3](#_Toc58319508)

[Structuri de date temporare 3](#_Toc58319509)

[Structuri de date de legătură 4](#_Toc58319510)

[Formatul fișierelor utilizate 4](#_Toc58319511)

[Descrierea setului de date 4](#_Toc58319512)

[Descrierea coloanelor 4](#_Toc58319513)

[Modelul architectural 6](#_Toc58319514)

[Arhitectura sistemului 6](#_Toc58319515)

[Sabloane arhitecturale utilizate 6](#_Toc58319516)

[Descrierea componentelor 7](#_Toc58319517)

[Restricțiile de implementare 9](#_Toc58319518)

[Elemente de testare 9](#_Toc58319519)

[Componente critice 9](#_Toc58319520)

[Alternative 9](#_Toc58319521)

# Scopul documentului

Acest document are rolul de a descrie detaliat şi complet soluţia proiectată pentru sistemul software: ,,Platformă pentru determinarea inferenței pentru pacienții suspecți de COVID”.

# Conţinutul documentului

Documentul este constituit din trei secțiuni mari. Acestea sunt:

* Modelul datelor (Data Design). În această secțiune sunt prezentate structurile de date esențiale în implementarea soluției și schema bazei de date ce stă la baza acesteia.
* Modelul arhitectural (Architectural Design). Această parte este dedicată arhitecturii sistemului.
* Elemente de testare (Testing Issues). Acest segment se ocupă de descrierea componentelor critice ale aplicației și de o descriere generală a potențialelor îmbunătățiri.

# Modelul datelor

## Structuri de date globale

Programul nostru se bazează pe formarea unui Dată Frame care să

dispună datele din fișierul CSV într-o formă abordabilă. Data Frame-ul a fost construit utilizând biblioteca pandas din Python. Utilizând această bibliotecă, a fost posibilă importarea fișierului de date.

## Structuri de date temporare

Nu au fost utilizate structuri temporale care influentează in mod

fundamental soluția.

## Structuri de date de legătură

Am utilizat un set de date alterat pentru a putea fi folosit la encodare.

Alterarea constă în eliminarea anumitor coloane din setul de date primit și adăugarea altor coloane encodate. Coloane următoare au foste eliminate din setul de date internare fiind considerate irelevante analizezi: ,,instituție sursa”, ,,dată internare”, ,,dată rezultat testare”, ,,dată debut simptome declarate”, ,,simptome raportate la internare”, ,,diagnostic și semne de internare”. Am uniformizat forma de exprimare a sexelor: ce conținea ,,F’’ a fost trecut ca ,, FEMININ’’, iar ce era cu ,,M’’ a fost trecut la ,,MASCULIN’’ astfel încât să putem face encodarea mai târziu. Marcarea aceasta a fost făcută în scopul de a pregăti setul de date pentru a fi encodat. Un exemplu este modifciarea coloanei de simptome utilizând regex\_lists.

## Formatul fișierelor utilizate

Fișierul CSV ce oferă toate datele utilize testării și antrenării soluției a fost pus la dispoziție de cerința enunțului și se află la acest link: ,,<https://drive.google.com/file/d/1ReYNWgio53dIlhz0eNO5ibC60N-7EurF/view>“

## Descrierea setului de date

### Descrierea coloanelor

* ,,instituția sursă” – Această coloană conține două tipuri de intrări, anume,,X’’ și ,,Y’’. Această coloană va fi eliminate întrucât ea nu este relevanță pentru construirea soluției.
* ,,sex’’ – Intrările acestei coloane au fost mai întâi înlocuite cu ,,FEMININ’’ și ,,MASCULIN’’ astfel încât să fie uniformizată coloana, ca mai apoi să fie encodate în 0 pentru ,,FEMININ’’, și 1, pentru ,,MASCULIN’’.
* ,, vârsta” – Această coloană este pentru identificarea vârstei subiectului, dar aceasta are o variație de intrări cum ar fi: vrastă nulă, vârsta exprimată în ani și luni, vârste peste 100, nou născut. În concluzie, aceasta coloana a trebuit să treacă printr-un process de uniformizare.
* ,,data debut simptome declarate” – A fost considerată irelevantă, deci a fost eliminată.
* ,,simptome declarate” – Această coloană a necesitat o procesare mai complexă. A trebuit să fie unifromizată ca apoi să poată fi encodată. Mai mult detalii se găsesc mai departe.
* ,,data internare” – A fost considerata irelevanta, deci a fost eliminata.
* ,,simptome raportate la internare” – A fost considerată irelevantă, deci a fost eliminată.
* ,,diagnostic și semne de internare” – A fost considerată irelevantă, deci a fost eliminată.
* ,,istoric de calatorie’’ – Această coloana a fost encodata, fiind marcate cu 0 intrările negative care nu ridică suspiciuni cu privire la o infectare, și 1 pentru intrările pozitive.
* ,,mijloacele de transport folosite’’ – Această coloana prezintă cazurile în care s-a utilizat mașînă personală sau mijloace de transporta in comun. A fost uniformizată sim ai apoi encodata cu 0 pentru intrările negative și 1 pentru intrările pozitive.
* ,,confirmare contact cu o persoană infectată’’ – Această coloana prezintă cazurile în care s-a utilizat mașînă personală sau mijloace de transporta in comun. A fost uniformizată sim ai apoi encodata cu 0 pentru intrările negative și 1 pentru intrările pozitive.
* ,,data rezultat testare’’ – A fost considerată irelevantă, deci a fost eliminată.
* ,,rezultat testare’’ – Această coloana a fost encodata pentru a putea face comparația dintre rezultatul testării din dată set și rezultatul testării ce a rezultat în urmă regresiei liniare.

# Modelul architectural

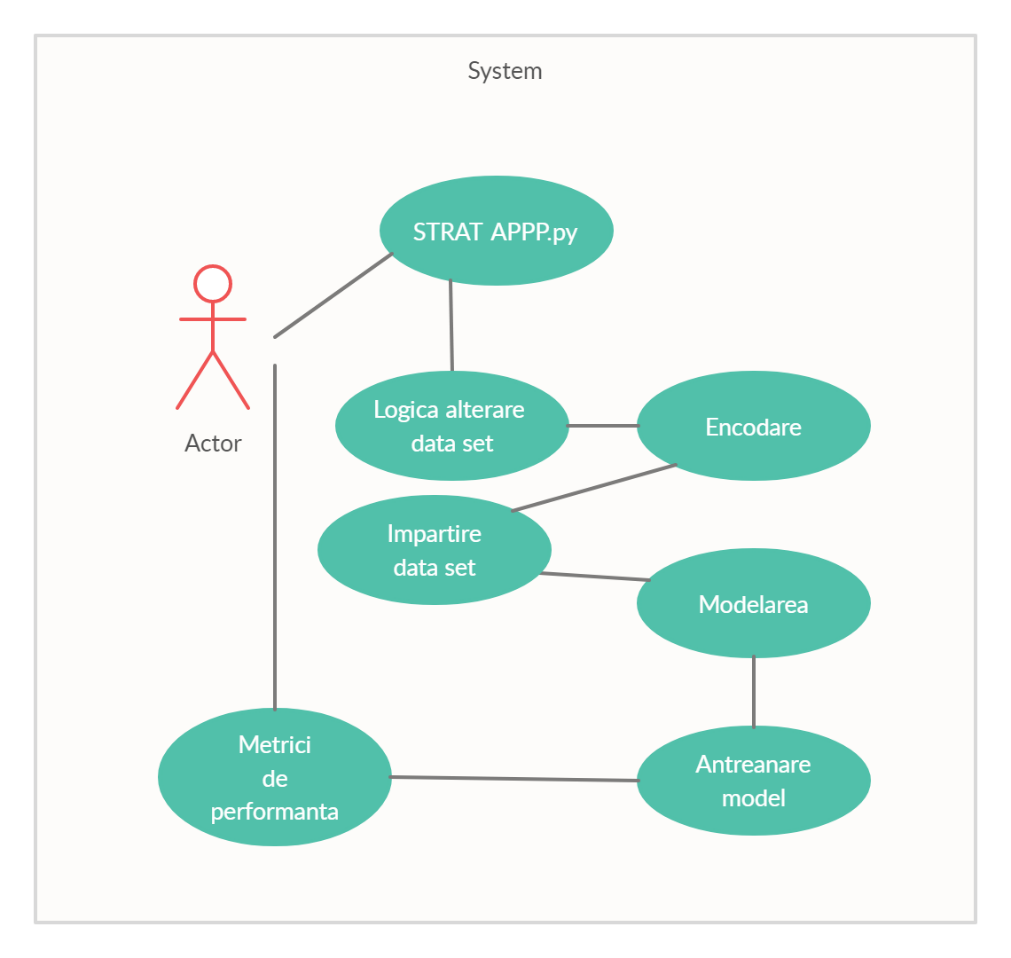
## Arhitectura sistemului

### Sabloane arhitecturale utilizate

În principiu, aplicația noastră se bazează pe utilizarea bibliotecilor

din Python, anume: pandas, sklearn, numpy, re, datetime. Sklearn a fost esențială pentru imparitre set de date, pentru construirea regresiei, encodare și pentru calcularea metricilor. Pandas pentru construirea DataFrame-ului. Numpy a fost folosită pentru lucrul cu tipuri de date. Re (Regular Expression operations) a fost util pentru preprocesarea datelor și anume pentru curățare

coloanelor.

Diagrama de arhitectura

## Descrierea componentelor

* START APP.py
* implemnatrea prog.py
* Logica alterare data set
* Constă în eliminarea coloanelor irelevante și în parsarea anumitor intrări pentru a putea fi encodate.Encodarea
* Encodarea este procesul de convertire a datelor dintr-o formă de tipul strîng într-o formă binară astfel încât datele să poată fi prelucrate. Specificațiile care nu ridică suspiciunea unei infectării cu virusul SARS-COV2 sunt notate cu 0, iar specificațiile ce pot conduce la un rezultat poztiv la testarea pentru virus sunt notate cu 1.
* Un exemplu pentru a detalia această alterarea pregătitoare encodării este următorul: intrările din coloana ,,istoric de călătorie” sunt marcate cu 1 în cazul subiecților care au călătorit și cu 0 în cazul celor care nu au călătorit.
* In cazul coloanei cu simptome, a fost nevoie de o prelucrare mai

complexă. Având în vedere faptul că pentru fiecare pacient erau trecute mai multe simptome declarate, a fost necesară parsarea acestei coloane prin transformarea fiecărei întrări în câte o lista de simptome. Mai departe, s-a aplicat un MultiLabelBinarizer care produce o nouă intrare intr-un nou DataFrame din fiecare simptom nou regăsit în listele de simptome. În aceste coloane nou apărute, se vor regăși valori binare: 1 pentru pacienții care au aceste simptome si 0 pentru cei ce nu le au.

* Împărțirea data set
* Împărțirea dată set-ului folosind un procent adecvat astfel încât să obținem o acuratețe cât mai mare: train\_test\_split(x\_data, y\_data, test\_size = 0.2).
* Modelarea
* Pentru partea de modelare am folosit modelul de Linear Regression din sklearn care va prezice pentru fiecare intrare rezultatul testului.
* Antrenare model
* Pentru acest pas folosim metoda de fit:

model.fit(x\_training\_data, y\_training\_data).

* Metrici de performanță
* Acestea se calculează pe baza match-ului dintre rezultatul regresiei și rezultatul din dată set. Se calculează: acuratețea, precizia, scor f1, AUC, matricea de confuzie, rapel.

# Restricțiile de implementare

Restrctiile de implementare sunt date de formă inițială a dată set-ului. A

fost nevoie de encodare astfel încât să putem modela și construi soluția, nefiind posibil acest lucru pe string-urile din data set.

# Elemente de testare

## Componente critice

Este foarte important modul de alegerea a coloanelor în scopul

construirii modelului, procentul de împărțire a dată set-ului. Este essential de analizat problematica echilibrării dată set-ului în cee ace privește label-urile astfel încât să nu se ajungă la overfitting.

## Alternative

O alternativă pentru implementarea soluției într-o manieră mai

performantă ar fi fost aceea de a utiliza un data balancer. (smote – un utilitar pentru dată balancer pentru problema unei clasificări a unui set de date imbalanced)