Heart disease

* **Scop** : o aplicație in R care ia datele pacientului ca intrări și prezice șansele prezenței

unei boli de inimă. Acest cod construieste o interfata grafica de utilizator (dashboard) in R utilizand pachetul "shiny".

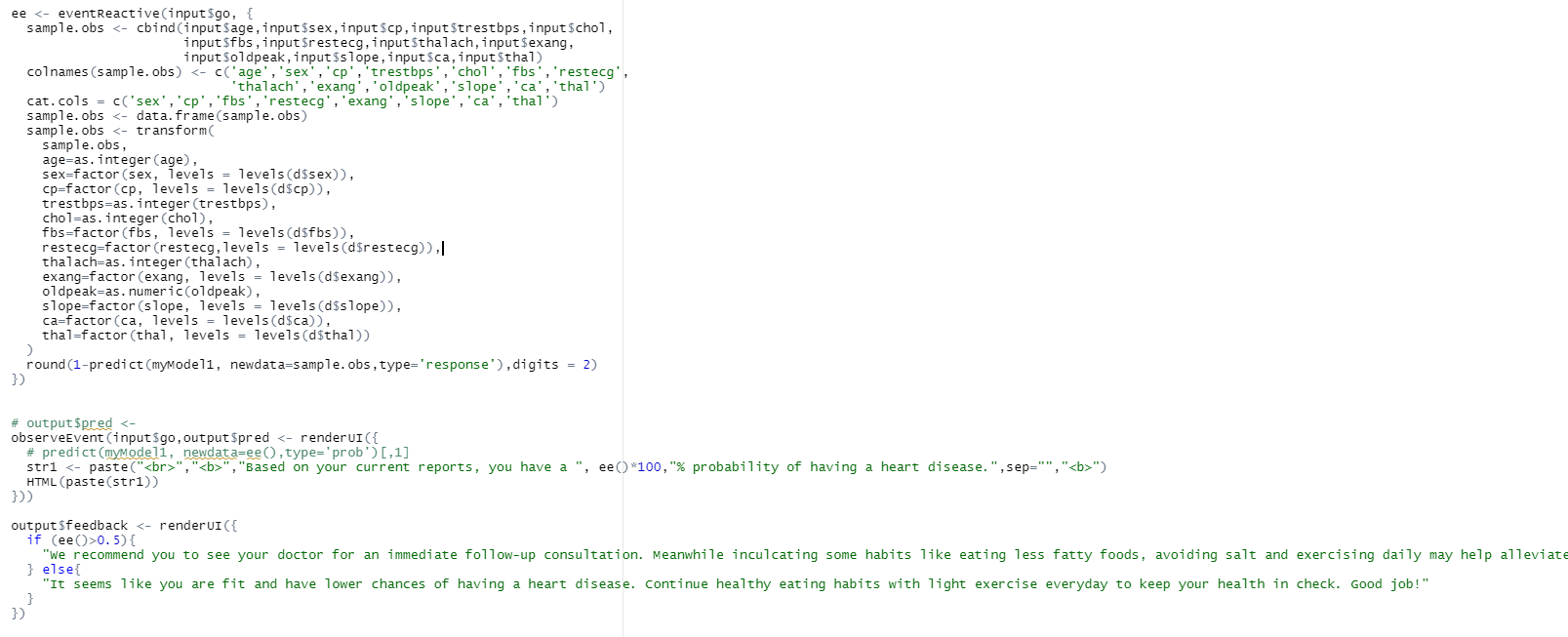
* **Componentele fisierului de intrare “processed\_cleveland.csv”:**

Aceste coloane reprezinta atribute sau caracteristici ale unui set de date care descriu pacientii cu probleme cardiace.

* 'age': varsta pacientului
* 'sex': genul pacientului (0 pentru femei si 1 pentru barbati)
* 'cp': tipul de durere toracica resimtita de pacient (valori posibile: 0, 1, 2, 3)
* 'trestbps': presiunea sangelui la pacient in timpul repausului (in mm Hg)
* 'chol': nivelul colesterolului seric al pacientului (in mg/dL)
* 'fbs': nivelul de glucoza din sange dupa ce pacientul a fost in post (0 pentru valori

normale si 1 pentru valori crescute)

* 'restecg': rezultatul electrocardiogramei la pacient in repaus (valori posibile: 0, 1, 2)
* 'thalach': frecventa maxima a inimii pacientului atinsa dupa efort (in batai pe minut)
* 'exang': prezenta sau absenta durerilor toracice legate de efort fizic (0 pentru absenta si 1 pentru prezenta)
* 'oldpeak': depresia celui mai mare unda T resimtita in timpul efortului fizic (in mm)
* 'slope': panta celui mai mare unda T resimtita in timpul efortului fizic (valori posibile: 1, 2, 3)
* 'ca': numarul de vase principale care au arata semne de ateroscleroza pe imagini (0-3)
* 'thal': rezultatul testului de stres pentru hipertensiunea arteriala (valori posibile: 3, 6, 7)
* 'target': diagnosticul final al pacientului (0 pentru absenta bolii cardiace si 1 pentru prezenta bolii cardiace)
* **Descriere cod:**
* Se citește un fișier CSV care conține datele cu numele de fișier "processed\_cleveland.csv" și se alocă la obiectul "d". Coloanele din fișier sunt denumite cu numele specificate în lista "c ('age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target')".
* Se modifică tipul de date pentru anumite coloane, cum ar fi "ca" și "thal" la integer.
* Se elimină rândurile care conțin valori nule în coloanele "ca" și "thalach".
* Se face transformarea coloanei "target" astfel încât să aibă doar valori 0 sau 1, și să fie tip de date factor.
* Se elimină valorile nule din setul de date.
* Se împarte setul de date în set de antrenament și set de testare cu o proporție de 75% pentru setul de antrenament și 25% pentru setul de testare.
* Se construiește un model de regresie logistică (GLM) pentru prezicerea coloanei "y" folosind setul de antrenament.
* Construim dataset-ul pentru EDA prin citirea datelor din fisierul "processed.cleveland.data", atribuirea de nume pentru fiecare coloana si transformarea datelor astfel incat sa devina mai usor de inteles si utilizat in analize.
* Acest lucru se realizeaza prin crearea de noi coloane care contin informatii despre fiecare pacient, precum:
* • Daca au sau nu afectiunea inimii (target)
* • Rata zaharului din sange (fbs\_range)
* • Genul pacientului (Gender)
* • Daca au angina in timpul exercitiilor (exercise\_angina)
* • Forma pantei segmentului ST (slope\_segment)
* • Raportul electrocardiografic (restecg\_report)
* • Tipul durerii toracice (Chest\_Pain\_type)
* • Raportul testului thal (thal\_report)
* • Numarul de artere care au fost semnalate ca fiind blocate (ca\_number)
* • In final, se afiseaza primele randuri ale datelor prelucrate folosind functia "head()".
* Se creeaza o variabila binomiala care ne ajuta sa analizam daca persoana cu datele date este predispusa la boli cardiovasculare.
* Soluția noastră are la baza funcția GLM(), familie = “binomial”.
* GLM, acronim al Generalized Linear Models, este folosit pentru a modela relația dintre o valoare care se dorește aflata (în cazul nostru dacă persoana este predispusă la maladii cardiace) și Combinația liniara a vectorului de variabile date pentru a antrena modelul ( în fișierul “processed\_cleveland.csv“)
* Modelul de Regresie Binomiala este caracterizat de variabila dependenta Y( variabila dependenta deoarece tine cont de datele prin care este antrenata, variabilele dependente se schimba ca rezultat al manipulării valorilor independente) care este o variabila aleatoare discreta.
* Vom face un dashboard care va avea un header cu titlul "Heart Disease Classification" si un sidebar cu 4 itemi de meniu: "EDA-1", "EDA-2", "EDA-3", si "Prediction". Body-ul dashboard-ului va avea 4 tab-uri, fiecare corespunzator unui item din meniul din sidebar. Fiecare tab va conține o serie de elemente grafice (specificate prin "frow1", "frow2", etc.), acestea fiind probabil definite in alta parte in cod.
* **EDA-1**
* Se creează două "randuri fluide" (fluidRow), fiecare cu două "chenare" (box) în interior. Fiecare chenar conține o intrare de utilizator și un grafic.
* Primul chenar din primul rând conține o intrare numerică pentru vârstă și un grafic de densitate care arată distribuția vârstei în raport cu boala de inimă.
* A l doilea chenar din primul rând conține un grup de casete de selectare pentru gen și un grafic de bară care arată distribuția genului în raport cu boala de inimă.
* La fel, al doilea rand conține o intrare slider și un grafic de densitate pentru "trestbps" și un grup de casete de selectare pentru "ChestPaintype" și un grafic de bară.
* Fiecare grafic este generat din datele din "df", un dataframe R, și afișat prin intermediul "plotOutput".
* Este utilizat un mecanism de reactivitate pentru a oferi actualizări dinamice ale datelor în funcție de alegerea utilizatorului în intrările de utilizator.
* **EDA-2**
* Acest cod crează două secțiuni numite frow3 și frow4 în aplicația Shiny. Fiecare secțiune conține două box-uri, fiecare cu câte un grafic și un control de intrare asociat. Controlul de intrare poate fi un slider sau un grup de căsuțe de selectare sau un câmp numeric.
* Fiecare box are un grafic legat de un output specific numit, de exemplu, "Chol", "Thalach", "fbs\_range", "restecg\_report". Aceste grafice sunt create utilizând biblioteca ggplot2 și sunt actualizate în funcție de intrarea utilizatorului.
* Există, de asemenea, blocuri de cod "reactive" care filtrează setul de date "df" în funcție de intrarea utilizatorului. Acestea sunt apoi utilizate pentru a crea graficele din fiecare box. De exemplu, blocul "data3" filtrează datele în funcție de valoarea selectată pentru "trestbps" prin intermediul controlului de intrare slider.
* În final, graficele sunt renderizate prin intermediul comenzii "renderPlot" și afișate în aplicația Shiny.
* **EDA-3**
* Primele două "boxuri" conțin grafice care prezintă relația dintre boala cardiovasculară și angina cu efort (Heart Disease vs Exercise Angina) respectiv relația dintre boala cardiovasculară și gradul de inclinare al segmentului (Heart Disease vs Slope Segment).
* Utilizatorii pot selecta valorile pentru "angina cu efort" și "grad de inclinare" folosind casetele de selectare. Al treilea "box" conține un grafic care prezintă relația dintre boala cardiovasculară și "oldpeak" și un slider pentru a selecta intervalul pentru "oldpeak".
* Ultimul "box" conține un grafic care prezintă relația dintre boala cardiovasculară și numărul de artere (ca\_number) și un element de intrare numerică pentru a selecta valoarea pentru "thalach".
* **Prediction**
* Se defineste un obiect "frow8", care este un rand fluid (un rand cu elemente care se pot adapta in functie de dimensiunea ecranului) in Shiny.
* In acest rand, se definesc diverse elemente de intrare utilizator, cum ar fi câmpuri de intrare numerice, butoane radio, buton de acțiune și output-uri HTML.
* Aceste elemente de intrare vor fi folosite pentru a obține informații despre o persoană (varsta, sexul, tipul de durere toracică, tensiunea arterială la repaus, nivelul colesterolului, etc.) și apoi vor fi utilizate pentru a face o predicție asupra prezenței bolii cardiace.
* La apăsarea butonului "Get Prediction", se va afișa rezultatul predicției într-un output HTML.
* Input-urile utilizatorului sunt preluate prin intermediul obiectului "input", iar output-ul este afisat prin intermediul obiectului "output".
* De exemplu, cand utilizatorul selecteaza un anumit tip de durere toracica, acesta este preluat prin intermediul "input$ChestPaintype" si apoi este utilizat in subset() pentru a filtra datele din "df".
* Graficul final este apoi creat prin intermediul ggplot() si afisat prin intermediul renderPlot(). Acest proces este repetat pentru fiecare caracteristica de boala de inima.
* **Grafice**
* Aplicatia utilizeaza un data frame numit "df" pentru a genera 12 grafice diferite care ilustreaza relatiile dintre diferitele caracteristici ale bolilor de inima si riscul de aparitie a acestora. Fiecare grafic este generat prin intermediul unui reactive() care filtreaza datele din "df" in functie de input-ul utilizatorului. Ulterior, fiecare grafic este creat utilizand functia ggplot() din pachetul ggplot2 si este afisat prin intermediul unui renderPlot().
* **Predictii**

****

* Blocul de cod se concentrează asupra rezultatelor predicției probabilității unei boli de inimă.
* observeEvent(input$go) este un eveniment reactiv care activează output$pred, care este un element de interfeță utilizator definit prin apelarea funcției renderUI. Acest eveniment poate fi activat prin apăsarea unui buton sau prin alte acțiuni specifice.
* În output$pred, se utilizează funcția paste() pentru a concatena mai multe string-uri și se apelază funcția HTML() pentru a afișa string-ul rezultat ca cod HTML. Funcția ee() returnează probabilitatea boli de inimă.
* În output$feedback, se utilizează un bloc if-else pentru a afișa un feedback diferit în funcție de probabilitatea boli de inimă. Dacă probabilitatea este mai mare de 0,5, se recomandă pacientului să se consulte cu un medic, altfel se afișează un mesaj pozitiv.