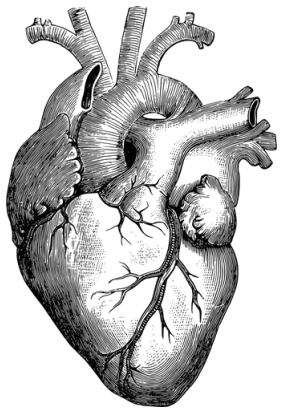
Matúš Balogh 25%
Myroslava Hrechyn 25%
Jana Viktória Kováčiková 25%
Rudolf Nosek 25%

Predikcia srdcového ochorenia

Logistická regresia pomocou kvázinewtonovských metód

11.05.2022



V tomto projekte odhadujeme pravdepodobnosť prítomnosti srdcového ochorenia u pacienta na základe výsledkov jeho vyšetrení. K dispozícii máme súbor heart.csv s údajmi o $m = 253$ pacientoch, ktorí podstúpili krvné testy. Každý riadok prislúcha práve jednému pacientovi. Posledný stĺpec obsahuje binárnu premennú v, ktorá nadobúda hodnotu 1, ak pacientovi bolo diagnostikované nejaké srdcové ochorenie, inak nadobúda hodnotu 0. Ostatné stĺpce (prvý až piaty) obsahujú premenné $u_1, u_2, ..., u_5$, ktoré udávajú vek a namerané hodnoty pozorovaných ukazovateľov (pokojový krvný tlak, cholesterol, maximálny krvný tlak, zmena na kardiograme).

Na modelovanie pravdepodobnosti použijeme logistickú regresiu. Logistická funkcia má tvar

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Zaveďme vektor parametrov logistickej regresie $x = (x_0, x_1, ..., x_5)^T \in \mathbb{R}^6$ a označme $u = (1, u_1, ..., u_5)^T \in \mathbb{R}^6$. Potom sa bude hodnota logistickej funkcie určovať v bodoch

$$x^T u = x_0 + x_1 u_1 + ... + x_n u_n.$$

Hodnota

$$g(x^T u) = \frac{1}{1 + e^{-x^T u}} \tag{1}$$

sa potom interpretuje ako pravdepodobnosť toho, že pacient vo veku u_1 s nameranými hodnotami $u_2, ..., u_5$ má srdcové ochorenie, teda $g(x^T u) = P(v = 1 | u_1, ..., u_5)$.
Našou úlohou je odhadnúť parametre logistickej regresie $x \in \mathbb{R}^6$ na základe údajov v a u. To vedie k optimalizačnej úlohe

$$Min \left\{ J(x) = -\sum_{i=1}^{m} v^i \, ln\left(g(x^T u^i)\right) + (1 - v^i) \, ln\left(1 - g(x^T u^i)\right) \mid x \in \mathbb{R}^6 \right\}, \tag{2}$$

kde $u^i = (1, u_1^i, ..., u_5^i)^T$. Všimnime si, že predpis účelovej funkcie J(x) je zvolený tak, aby sa penalizovala nízka pravdepodobnosť $g(x^T u^i)$ toho, že pacient má srdcové ochorenie, ak ho naozaj má ($v^i = 1$) a vysoká pravdepodobnosť $g(x^T u^i)$ toho, že pacient má srdcové ochorenie, ak ho v skutočnosti nemá ($v^i = 0$).

a) Chceme ukázať, že predpis funkcie J(x) v úlohe (2) sa dá zjednodušiť na tvar

$$\boldsymbol{Min}\left\{\boldsymbol{J(x)=\sum_{i=1}^{m}(1-v^i)x^Tu^i+ln\left(1+e^{-x^Tu^i}\right)\ |\ x\in\mathbb{R}^6}\right\}. \tag{3}$$

Dosadením funkcie (1) do predpisu funkcie $J(x)$ v (2) dostaneme vzťah

$$J(x)=-\sum_{i=1}^{m}\left(v^i\ ln\left(\frac{1}{1+e^{-x^Tu^i}}\right)+(1-v^i)\ ln\left(1-\frac{1}{1+e^{-x^Tu^i}}\right)\right).$$

Členy v poslednej zátvorke upravíme na spoločného menovateľa:

$$\begin{aligned}J(x)&=-\sum_{i=1}^{m}\left(v^i\ ln\left(\frac{1}{1+e^{-x^Tu^i}}\right)+(1-v^i)\ ln\left(\frac{1+e^{-x^Tu^i}-1}{1+e^{-x^Tu^i}}\right)\right)\\&=-\sum_{i=1}^{m}\left(v^i\ ln\left(\frac{1}{1+e^{-x^Tu^i}}\right)+(1-v^i)\ ln\left(\frac{e^{-x^Tu^i}}{1+e^{-x^Tu^i}}\right)\right).\end{aligned}$$

Použijeme vzorec pre podiel logaritmu $ln(\frac{x}{y})=ln(x)-ln(y)$:

$$J(x)=-\sum_{i=1}^{m}\left(v^i\left(ln(1)-ln(1+e^{-x^Tu^i})\right)+(1-v^i)\left(ln(e^{-x^Tu^i})-ln(1+e^{-x^Tu^i})\right)\right).$$

Ďalej využijeme vzorec $ln(e^x)=x$ a vynecháme člen $ln(1)=0$. Dostávame tak

$$J(x)=-\sum_{i=1}^{m}\left(-v^i\ ln(1+e^{-x^Tu^i})+(1-v^i)\left(-x^Tu^i-ln(1+e^{-x^Tu^i})\right)\right).$$

Roznásobíme zátvorky

$$J(x)=-\sum_{i=1}^{m}\left(-v^i\ ln(1+e^{-x^Tu^i})-x^Tu^i-ln(1+e^{-x^Tu^i})+v^ix^Tu^i+v^i\ ln(1+e^{-x^Tu^i})\right),$$

a vidíme, že prvý a posledný člen sa vykrátia. Už len vyjmeme x^Tu^i pred zátvorku a mínus pred sumou presunieme do vnútra sumy:

$$J(x)=\sum_{i=1}^{m}\left(x^Tu^i\ (1-v^i)+ln(1+e^{-x^Tu^i})\right).$$

Získali sme teda požadovanú rovnosť

$$J(x) = -\sum_{i=1}^{m} v^i \; ln\left(g(x^T u^i)\right) + (1 - v^i) \; ln\left(1 - g(x^T u^i)\right) =$$
$$= \sum_{i=1}^{m} \left((1 - v^i) \; x^T u^i + ln(1 + e^{-x^T u^i})\right),$$

z čoho vyplýva

$$Min\left\{J(x) = -\sum_{i=1}^{m} v^i \; ln\left(g(x^T u^i)\right) + (1 - v^i) \; ln\left(1 - g(x^T u^i)\right) \mid x \in \mathbb{R}^6\right\} = \tag{2}$$
$$= Min\left\{J(x) = \sum_{i=1}^{m} (1 - v^i) x^T u^i + ln\left(1 + e^{-x^T u^i}\right) \mid x \in \mathbb{R}^6\right\}. \tag{3}$$

□

b) Vyjadrime prvky gradientu $\nabla J(x)$ účelovej funkcie $J(x)$ v tvare (3).

Vektory x^T a u^i vo funkcii $J(x)$ si rozpíšeme po prvkoch

$$J(x) = \sum_{i=1}^{m} \left((1 - v^i)\, x^T u^i + ln(1 + e^{-x^T u^i}) \right) =$$

$$= \sum_{i=1}^{m} (1 - v^i)(x_1 \cdot 1 + x_2 \cdot u_1^i + x_3 \cdot u_2^i + x_4 \cdot u_3^i + x_5 \cdot u_4^i + x_6 \cdot u_5^i) + ln(1 + e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i})$$

a vyjadríme jednotlivé parciálne derivácie podľa x_1 až x_6:

$$\frac{\partial J}{\partial x_1} = \sum_{i=1}^{m} (1 - v^i) \cdot 1 + \frac{1}{1 + e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i}} \cdot e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i} \cdot (-1)$$

$$\frac{\partial J}{\partial x_2} = \sum_{i=1}^{m} (1 - v^i) \cdot u_1^i + \frac{1}{1 + e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i}} \cdot e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i} \cdot (-u_1^i)$$

$$\frac{\partial J}{\partial x_2} = \sum_{i=1}^{m} (1 - v^i) \cdot u_2^i + \frac{1}{1 + e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i}} \cdot e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i} \cdot (-u_2^i)$$

...

$$\frac{\partial J}{\partial x_6} = \sum_{i=1}^{m} (1 - v^i) \cdot u_5^i + \frac{1}{1 + e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i}} \cdot e^{-x_1 \cdot 1 - x_2 \cdot u_1^i + \ldots - x_6 \cdot u_5^i} \cdot (-u_5^i)$$

To môžeme ešte poupravovať:

$$\begin{aligned}
\frac{\partial J}{\partial x_k} &= \sum_{i=1}^{m} \left((1 - v^i) \cdot u_{k-1}^i + \frac{1}{1 + e^{-x^T u^i}} \cdot e^{-x^T u^i} \cdot (-u_{k-1}^i) \right) = \\
&= \sum_{i=1}^{m} \left((1 - v^i) \cdot u_{k-1}^i + \frac{1}{1 + e^{-x^T u^i}} \cdot \frac{1}{e^{x^T u^i}} \cdot (-u_{k-1}^i) \right) = \\
&= \sum_{i=1}^{m} \left((1 - v^i) \cdot u_{k-1}^i - \frac{u_{k-1}^i}{e^{x^T u^i} + 1} \right) = \\
&= \sum_{i=1}^{m} u_{k-1}^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i} + 1} \right)
\end{aligned}$$

pre $k = 1, .., 6$, ak u_0^i definujeme ako 1.

Gradient $\nabla J(x)$ vyjadrený po prvkoch je teda:

$$\nabla J(x) = \begin{pmatrix} m - \sum_{i=1}^{m} \left(v^i + \frac{1}{e^{x^T u^i}+1} \right) \\ \sum_{i=1}^{m} u_1^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i}+1} \right) \\ \sum_{i=1}^{m} u_2^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i}+1} \right) \\ \sum_{i=1}^{m} u_3^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i}+1} \right) \\ \sum_{i=1}^{m} u_4^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i}+1} \right) \\ \sum_{i=1}^{m} u_5^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i}+1} \right) \end{pmatrix}.$$

Alternatívne sme mohli gradient vypočítať priamo, bez rozpisovania na parciálne derivácie:

$$\nabla J(x) = \sum_{i=1}^{m} \left((1 - v^i) \cdot u_1^i + \frac{1}{1 + e^{-x^T u^i}} \cdot e^{-x^T u^i} \cdot (-u^i) \right) = \sum_{i=1}^{m} u^i \cdot \left(1 - v^i - \frac{1}{e^{x^T u^i} + 1} \right).$$

Časti c) - g) sme naimplementovali v Pythone. Link na náš kód: kliknite sem
Implementácia kvázinewtonovských metód, backtrackingu a bisekcie sa nachádza v súbore optimizer.py Úlohy c) - e) sa nachádzajú v súbore test.ipynb a úlohy f), g) v súbore logistic_regression.ipynb.

Kvázinewtonovské metódy sú metódy voľnej optimalizácie inšpirované Newtonovou metódou, ktorá je z klasických metód síce najefektívnejšia, avšak algoritmus Newtonovej metódy môže byť výpočtovo náročný kvôli výpočtu Hessovej matice druhých parciálnych derivácií a následnému riešeniu sústavy n lineárnych rovníc

$$\left[\nabla^2 f(x^k)\right](x^{k+1} - x^k) = -\nabla f(x^k).$$

Myšlienka kvázinewtonovských metód spočíva v aproximácii inverznej Hessovej matice

$$H_k \approx \left[\nabla^2 f(x^k)\right]^{-1},$$

čím odpadne problém s riešením systému lineárnych rovníc - smer s_k sa totiž získa priamo ako $s_k = -H_k \nabla f(x^k)$. Aproximácia prebieha postupne a v každej iterácii sa "vylepšuje". Štartovaciu maticu H_0 spravidla aproximujeme identitou, a H_{k+1} spočítame pomocou symetrickej korekčnej matice ΔH_k ako

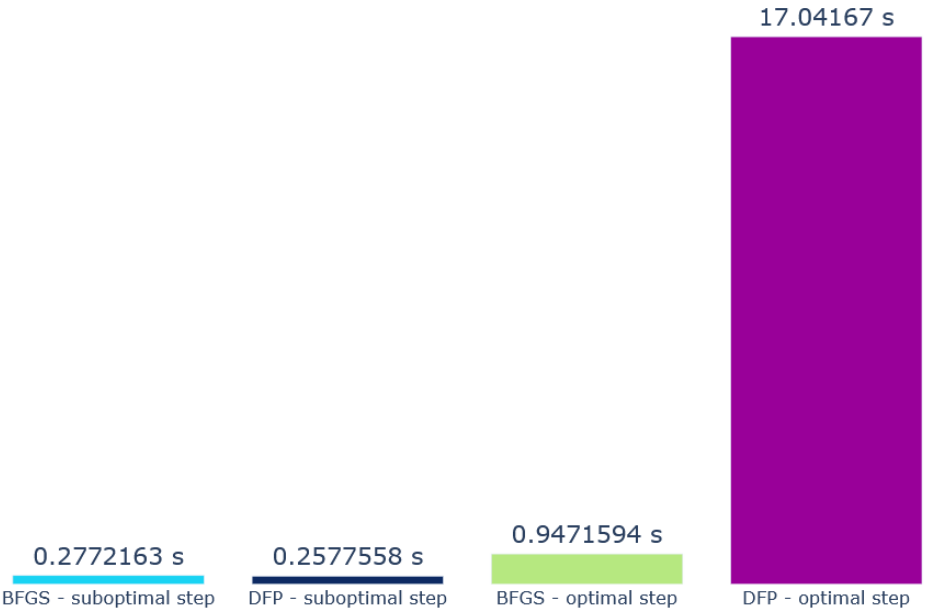
$$H_{k+1} = H_k + \Delta H_k.$$

Hlavnou výhodou kvázinewtonovských metód je teda ich rýchlosť.

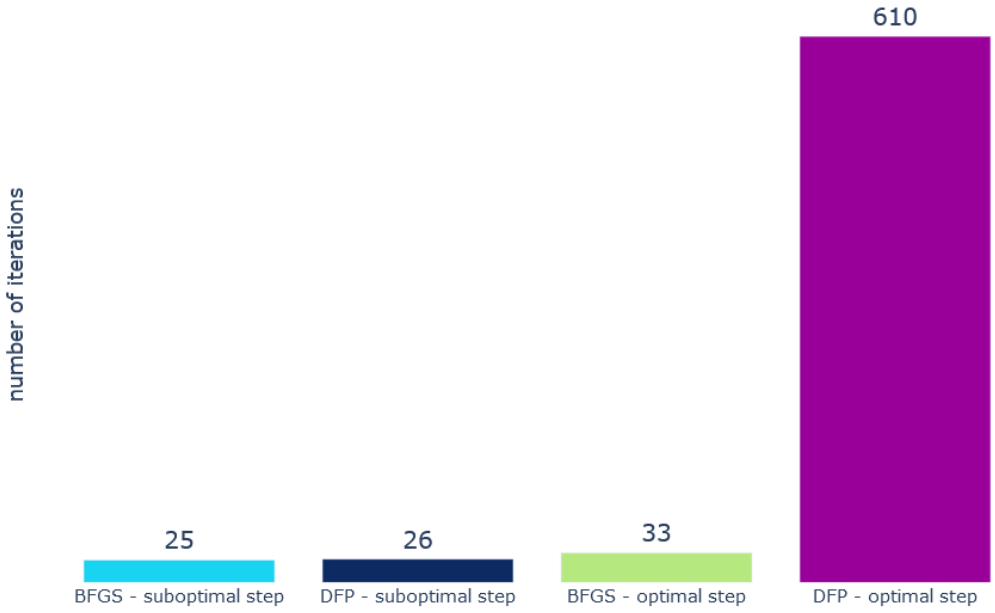
c) Úlohu (3) sme riešili pomocou BFGS metódy s približne optimálnou dĺžkou kroku a DFP metódy s približne optimálnou dĺžkou kroku. Na hľadanie približne optimálnej dĺžky kroku sme využili metódu backtracking (bez potreby odvodenia Hessovej matice). Využili sme gradient z časti b), štartovací krok sme zvolili $\boldsymbol{x_0 = (-2.28, 0, 0, 0, 0, 0)^T}$ a ako kritérium optimality sme použili $||\nabla \boldsymbol{J}(\boldsymbol{x^k})|| \leq \epsilon = \boldsymbol{10^{-3}}$.

d) Úlohu (3) sme riešili pomocou BFGS metódy s optimálnou dĺžkou kroku a DFP metódy s optimálnou dĺžkou kroku. Na hľadanie optimálnej dĺžky kroku sme využili metódu bisekcie. Volili sme vstupné parametre z časti c) a merali sme trvanie výpočtu.

Porovnanie výsledkov:

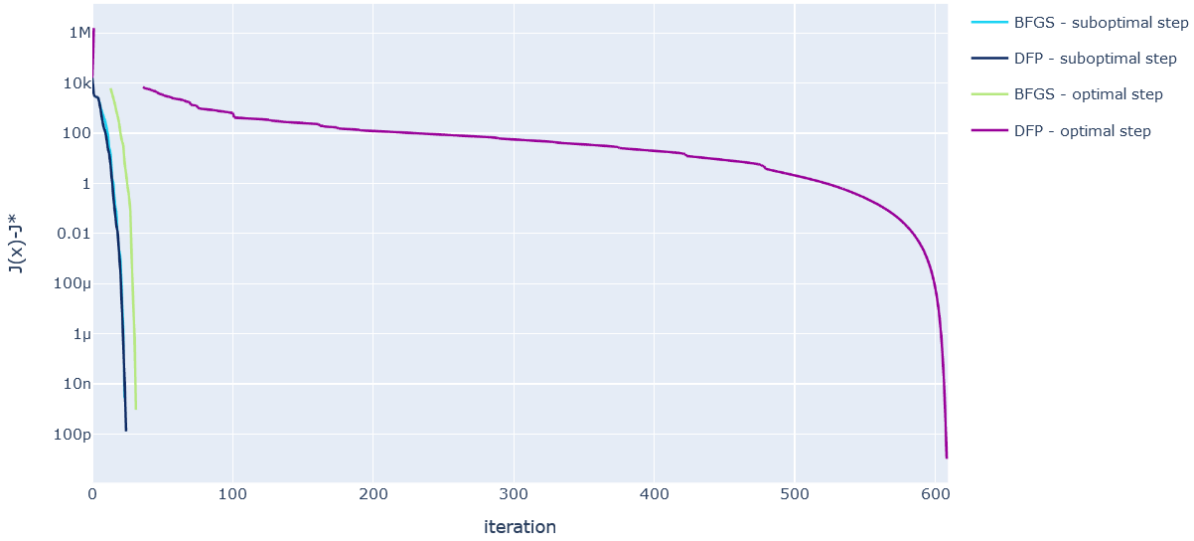


Porovnanie trvania výpočtu jednotlivých kvázinewtonovských metód

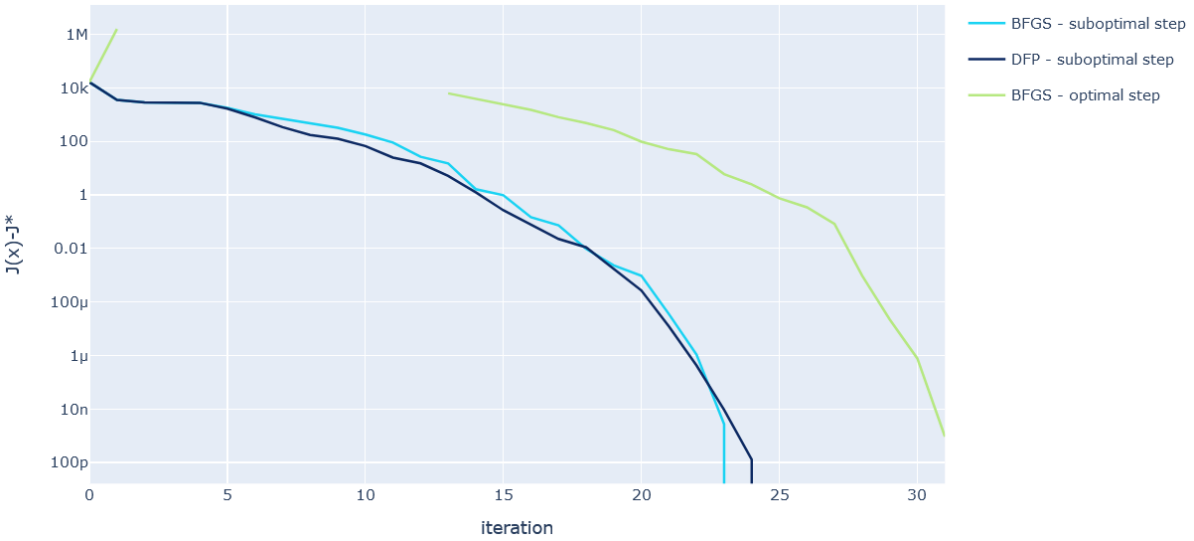


Porovnanie počtu iterácií jednotlivých kvázinewtonovských metód

e) Symbolom J^* označme nájdené ϵ-presné riešenie. Pre rôzne kvázinewtonovské metódy sme vykreslili do jedného obrázku grafy znázorňujúce vývoj hodnoty $J(x^k) - J^*$ s rastúcim číslom iterácie k. Na osi y sme použili logaritmickú mierku.



Rozdiel $J(x^k) - J^*$ pri jednotlivých kvázinewtonovských metódach



Pohľad zblízka - rozdiel $J(x^k) - J^*$ pri jednotlivých kvázinewtonovských metód (okrem DFP metódy s optimálnym krokom)

Zhrnutie porovnania metód

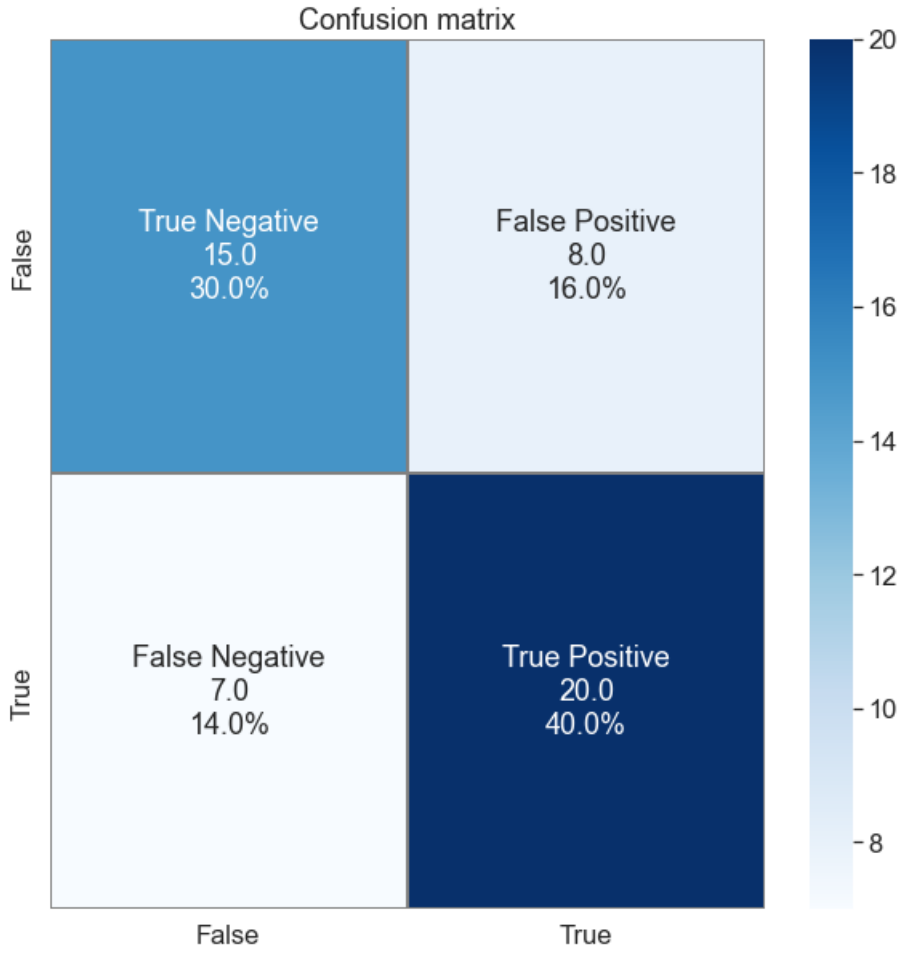
- **najrýchlejšia** bola metóda DFP s približne optimálnym krokom,

- **najlepšiu optimálnu hodnotu** mala DFP metóda s približne optimálnym krokom,

- **najmenší počet iterácií** potrebovala metóda BFGS s približne optimálnym krokom (no metóda DFP s pribl. opt. krokom skončila v jej tesnom závese s rozdielom 1 iterácie).

Na základe týchto kritérií považujeme metódu **DFP s približne optimálnym krokom za najvhodnejšiu** pre našu aplikáciu.

f) Riešením úlohy (3) sme odhadli model na binárnu klasifikáciu, t.j. ak pravdepodobnosť toho, že pacient má srdcové ochorenie, je aspoň 50 %, tak ho klasifikujeme ako chorého, inak ako zdravého. Na dátach zo súboru heart_test.csv, ktoré majú rovnakú štruktúru ako pôvodné dáta heart.csv, sme otestovali, ako tento model klasifikuje prítomnosť srdcového ochorenia na základe pozorovaných ukazovateľov $u_1, ..., u_5$. Použili sme pri tom metódu DFP s približne optimálnou dĺžkou kroku, nakoľko túto metódu sme vyhodnotili v predošlej časti pre danú úlohu za najlepšiu. Zaujímalo nás, akú časť pacientov klasifikuje model správne.

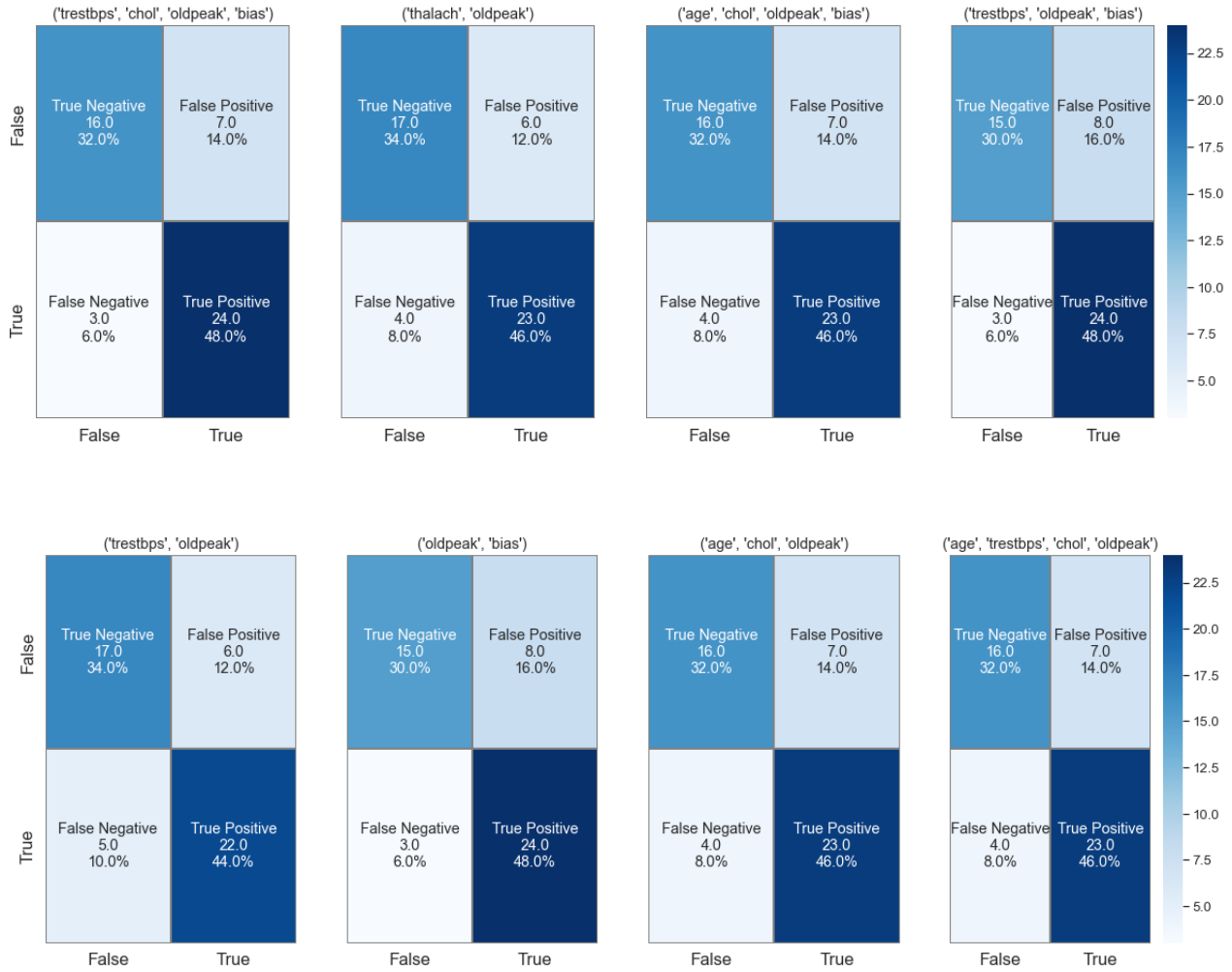


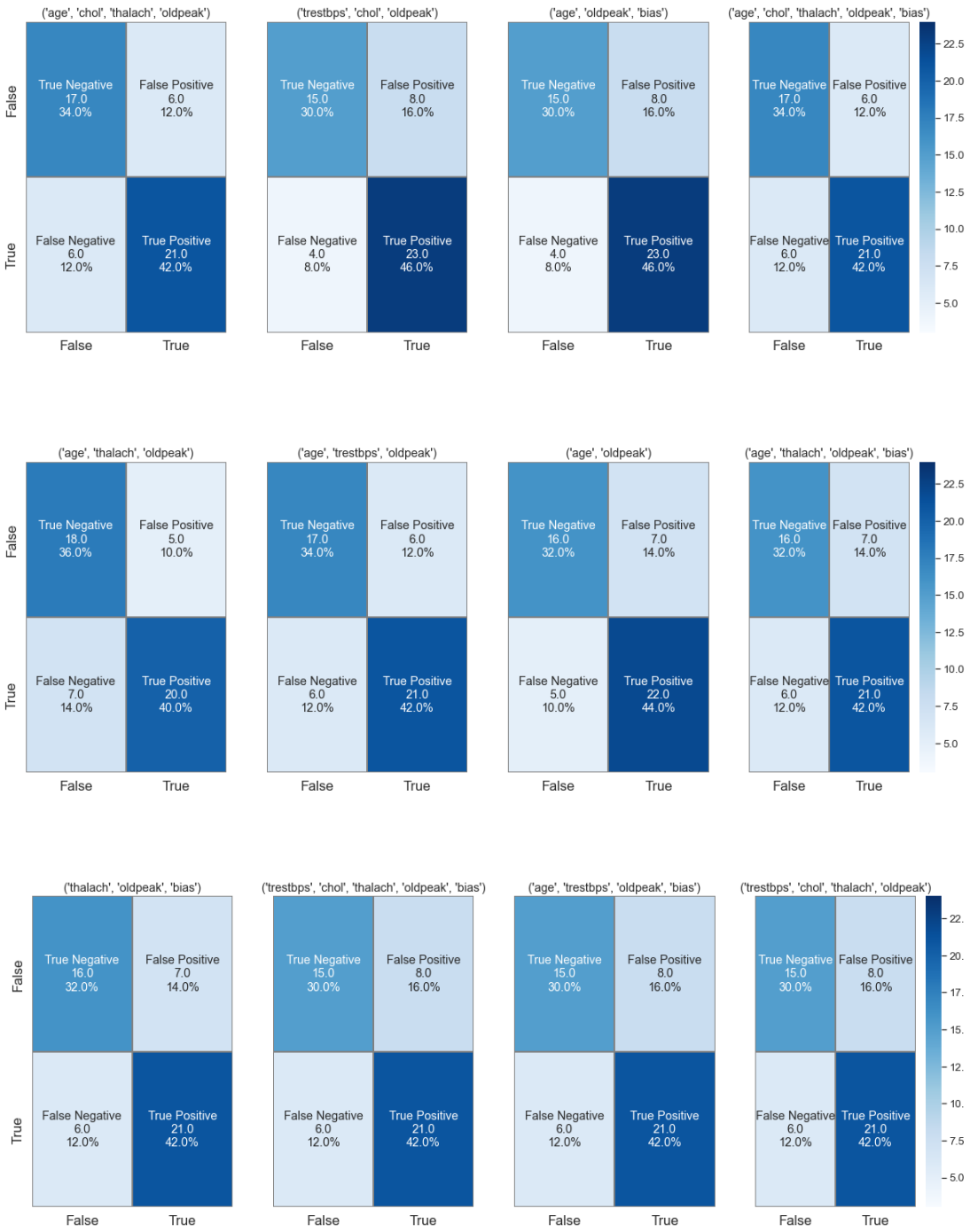
Výsledky modelu, metóda DFS s pribl. optimálnym krokom

Model na testovacích dátach vykazuje **senzitivitu 74.1%** a **špecificitu 65.2%**. V 16% prípadov bol falošne pozitívny a v 14% prípadov falošne negatívny. Model teda klasifikuje **70%** pacientov správne.

g) Skúšali sme projekt modifikovať 3 rôznymi spôsobmi.

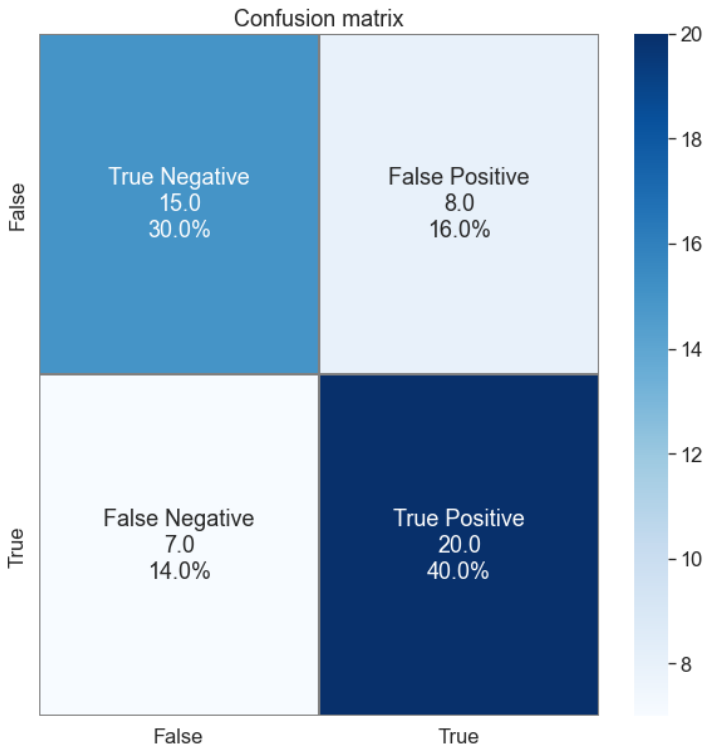
1. Vyskúšali sme namiesto celých dát vziať iba **vhodne zvolené kombinácie stĺpcov** a modelovali sme na základe týchto kombinácií. Najlepšie výsledky sme dostali pri kombinácii nasledovných 4 stĺpcov: pokojový krvný tlak, cholesterol, zmena na kardiograme a bias (vynechali sme teda maximálny krvný tlak). Pri takto zvolených stĺpcoch nám vyšli hodnoty: 32% skutočná negativita, 48% skutočná pozitivita, 6% falošná negativita a 14% falošná pozitivita, viď prvý z obrázkov v ľavom hornom rohu. Celkovo tak tento model vyhodnotil **80%** prípadov správne.
Ďalej sme si všimli, že všetky kombinácie, ktoré dávali lepší výsledok ako model na pôvodných dátach (všetkých stĺpcoch), obsahovali stĺpec **"oldpeak"**, čiže ukazovateľ "zmena na kardiograme".





Výsledky modelov s vhodne zvolenými kombináciami regresorov, ktoré klasifikovali viac pacientov správne ako pôvodný model v časti f.

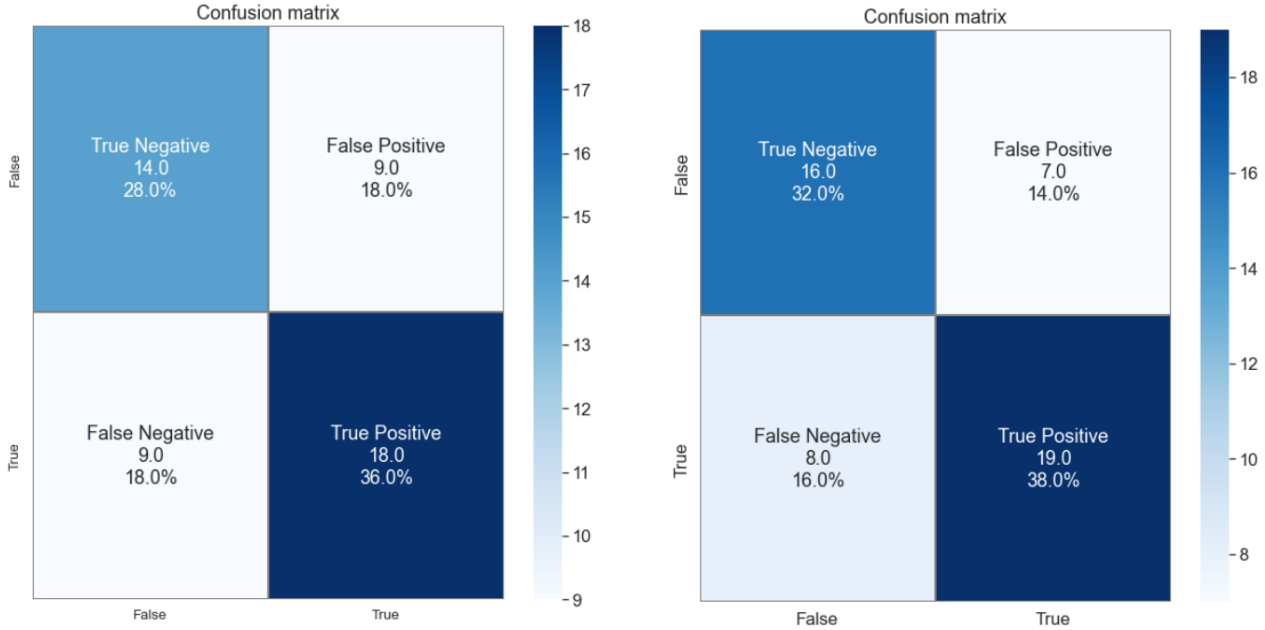
2. Dáta sme **normalizovali** a model sme skúsili použiť na takto poupravených dátach.



Výsledky modelu na normalizovaných dátach

Ako môžeme vidieť na obrázku, model použitý na normalizovaných dátach nám dal rovnaký výsledok ako model na nenormalizovaných dátach v časti f). Normalizácia teda v našom prípade predikciu **nezlepšila**.

3. Skúsili sme **poupraviť účelovú funkciu** $J(x)$ tak, že sme do jej predpisu pridali sumu štvorcov $x^T x$, v snahe čo najviac minimalizovať koeficienty. Následne sme model použili na pôvodných dátach, ako aj na normalizovaných dátach.



(a) pôvodné dáta (b) normalizované dáta

Výsledky modelu s poupravenou účelovou funkciou

Oproti pôvodnému modelu však nevidíme zlepšenie - kvalita klasifikácie tohto modelu (predovšetkým na pôvodných dátach) bola dokonca horšia než kvalita klasifikácie modelu s pôvodnou účelovou funkciou. Na normalizovaných dátach klasifikoval tento model síce rovnaké percento pacientov správne, avšak vykazoval väčšie množstvo falošne negatívnych výsledkov, čo v súvislosti s predikciou srdcového ochorenia môžeme považovať za nevýhodu.

Z našich vyskúšaných modifikácií teda najlepšie fungoval model použitý na 4 stĺpcoch, ktorý je popísaný v časti 1.

Zdroje

https://www.hiclipart.com/free-transparent-background-png-clipart-dvinm/download
https://plotly.com/python/bar-charts/
učebnica [Hamala,Trnovska]NelinearneProgramovnie.pdf
doplnkový materiál MVOtextkprednaske09.pdf
a zadanie úlohy LogistRegresiaKardioQNM.pdf.