FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Vizualizace chování algoritmů strojového učení po doplnění chybějících dat
Student:	Naďa Fučelová
Vedoucí:	Ing. Magda Friedjungová, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Umělá inteligence 2021
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Datové sady se v praxi často potýkají s chybějícími daty. V některých případech lze chybějící hodnoty odstranit, v jiných je naopak nezbytné chybějící data nějakým způsobem doplnit, zrekonstruovat. Pro doplnění chybějících dat existuje několik způsobů, ať už jde o doplnění pouze průměrem daného příznaku nebo pomocí metod jako je kNN, MICE či využití generativních modelů jako VAE nebo GAN. Cílem této práce je vyvinout systém pro vizualizaci chování vybraných modelů strojového učení vzhledem k chybějícím datům ve vybraných datových sadách a jejich doplnění různými metodami. Modely by měly být schopné rychlého natrénování na nových datech tak, aby byl efekt aplikace jednotlivých metod ihned pozorovatelný. Výsledný systém by měl být snadno rozšiřitelný o nové způsoby doplnění chybějících dat a modely strojového učení.

Pokyny k vypracování:
1) Proveďte rešerši metod pro doplňování chybějících dat, zaměřte se na data tabulární (tzn. ne obrazová).
2) Na základě provedené rešerše rozhodněte, které metody pro doplnění chybějících dat implementujete. Zvolte minimálně čtyři metody, pro implementaci lze využít i existující nástroje. Svoji volbu zdůvodněte.
3) Navrhněte a implementujte systém, který umožní:
- Nahrání datové sady uživatelem.
- Prohlížení jednotlivých řádků a sloupců datové sady, doplněné o popisné statistiky jednotlivých příznaků včetně poměru chybějících dat.

Elektronicky schválil/a Ing. Karel Klouda, Ph.D. dne 20. června 2023 v Praze.

- Výběr, natrénování a prezentace klasifikačního/regresního modelu na neupraveném datasetu.
- Výběr z vámi implementovaných metod pro doplnění chybějících dat, následně doplnění dat zvolenou metodou.
- Přetrénování klasifikačního/regresního modelu na doplněném datasetu, prezentace výsledků (např. změna v klasifikační přesnosti/MSE).

4) Systém experimentálně otestujte na třech datových sadách. Výsledky diskutujte.

Veškeré kroky pečlivě popište v písemné části práce. Systém navrhněte tak, aby byl dále rozšiřitelný o další metody.

Elektronicky schválil/a Ing. Karel Klouda, Ph.D. dne 20. června 2023 v Praze.

Bakalárska práca

VIZUALIZACE CHOVÁNÍ ALGORITMŮ STROJOVÉHO UČENÍ PO DOPLNĚNÍ CHYBĚJÍCÍCH DAT

Naďa Fučelová

Fakulta informačních technologií
Katedra aplikované matematiky
Vedúca: Ing. Magda Friedjungová, Ph.D.
12. mája 2024

Obsah

Zoznam obrázkov

Chcela by som poďakovať Ing. Magde Friedjungovej, Ph.D. za ochotu viesť moju prácu, za čas strávený konzultáciami a v neposlednom rade za poskytnuté cenné rady. Ďalej by som chcela poďakovať rodine a kamarátom za podporu počas celej doby štúdia.

Vyhlásenie

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Prahe dňa 12. mája 2024

Abstrakt

Dátové sady tvoria neoddeliteľnú súčasť strojového učenia, zohrávajú kľúčovú rolu pri trénovaní modelov a v konečnom dôsledku formujú ich predikcie. V reálnych dátových sadách však často niektoré hodnoty chýbajú, čo môže spôsobovať širokú škálu problémov od zníženej presnosti až po neschopnosť niektoré algoritmy vôbec použiť. Táto práca sa zaoberá skúmaním metód dopĺňania chýbajúcich hodnôt v tabulárnych dátových sadách a ich vplyvu na vybrané algoritmy strojového učenia. Výsledkom je webová aplikácia, ktorá má za cieľ uľahčiť pochopenie tejto problematiky. Aplikácia umožňuje na používateľom nahranú dátovú sadu aplikovať vybrané metódy a následne pozorovať ich vplyv na zvolený algoritmus strojového učenia prostredníctvom vizualizácií. Práca ďalej obsahuje niekoľko praktických ukážok použitia aplikácie na známych dátových sadách sprevádzaných diskusiou dosiahnutých výsledkov.

Kľúčové slová imputácia dát, webová aplikácia, strojové učenie, MICE, kNN, GAIN, MissForest

Abstract

Datasets are an integral part of machine learning, playing a key role in the training of models and shaping their predictions. However, real-world datasets often contain missing values, which can cause a wide range of problems from reducing their accuracy to the inability to apply some algorithms at all. This thesis examines data imputation techniques in tabular datasets and their impact on selected machine learning algorithms. The result of this thesis is a web application that aims to provide a better understanding of the given issue. The application allows users to apply selected techniques to the uploaded dataset and observe their effects on chosen machine learning algorithms through visualizations. Additionally, the thesis includes several practical examples of application use on common datasets, accompanied by a discussion of the achieved results.

Keywords data imputation, web application, machine learning, MICE, kNN, GAIN, MissForest

Zoznam skratiek

AUC	Plocha pod krivkou operačnej charakteristiky prijímača
GAIN	Generatívne súperiace imputačné siete
kNN	K-najbližších susedov
MAR	Chýbajúce náhodne
MCAR	Chýbajúce úplne náhodne
MICE	Viacrozmerná imputácia pomocou zreťazených rovníc
MissForest	Imputácia pomocou náhodného lesa
MissPALasso	Algoritmus striedania vzoru chýbania Lasso
MNAR	Nechýbajúce náhodne
MSE	Stredná kvadratická chyba
RMSE	Odmocnina strednej kvadratickej chyby
ROC	Operačná charakteristika prijímača
SVM	Metóda podporných vektorov
VAE	Variačný autoenkodér
WGAIN	Wassersteinova modifikácia GAIN

Úvod

Žijeme vo svete, ktorý býva označovaný za svet riadený dátami. Rôzne strategické rozhodnutia sú často prijímané na základe analýzy a interpretácie dostupných údajov. Dobrým príkladom je sektor dopravy, kde sa typicky sleduje zaťaženie spojov a na základe toho dochádza k posilneniu, alebo naopak k oslabeniu niektorých liniek. Obdobných príkladov však možno nájsť hneď niekoľko v takmer každom odvetví. Z tohto dôvodu je prvoradá spoľahlivosť dát a čo najmenšia odchýlka od reality. Nízka kvalita údajov môže zapríčiniť nespravodlivú automatizovanú voľbu, ktorá vedie až k ekonomickým stratám, alebo má iné negatívne spoločenské dopady [1]. Kvalitu však do značnej miery komplikuje prítomnosť chýbajúcich hodnôt, ktoré sú ich bežnou súčasťou.

Chýbajúce hodnoty môžu vzniknúť vo všetkých fázach životného cyklu dát, vrátane zberania, ukladania, či prenosu. Príčiny sú rôzne a nedajú sa úplne eliminovať. Nie je totiž napríklad možné donútiť respondenta odpovedať na všetky otázky a rovnako sa nedá zabrániť technickým chybám nástrojov, ktoré merajú údaje.

Pri práci s nekompletnou dátovou sadou možno zvoliť jeden z troch prístupov: dáta nijak neupravovať, odstrániť chýbajúce hodnoty, alebo ich naopak doplniť. Vhodnosť voľby závisí od viacerých faktorov, ako je napríklad počet chýbajúcich hodnôt, príčina ich vzniku, riešená úloha, alebo dostupné výpočtové kapacity. Metód odstránenia aj doplnenia dát je niekoľko, pričom nové spôsoby riešenia tejto problematiky sa stále objavujú.

Dáta však nie sú využívané len pre štatistické analýzy, no tvoria tiež základ algoritmov strojového učenia. Hovorí sa, že algoritmy sú len tak dobré, ako dáta, na ktorých sú trénované [2]. Práve vďaka nim totiž získavajú poznatky používané pri predikciách. V dôsledku doplnenia hodnôt do dátovej sady tak môže dôjsť k zmenám ich presnosti. Dôraz práce je kladený práve na pozorovanie správania algoritmov pred a po doplnení.

Teoretická časť obsahuje predstavenie problematiky chýbajúcich hodnôt, vysvetlenie mechanizmov ich vzniku a ďalej sa ponára do súčasných metód ich dopĺňania v tabulárnych dátových sadách.

Na rešerš nadväzuje praktická časť. Zaoberá sa hlavným cieľom tejto práce, ktorým je implementácia systému. Ten umožňuje užívateľovi doplniť chýbajúce hodnoty pomocou celkom piatich metód a na takto doplnených sadách natrénovať algoritmy strojového učenia. Pri dopĺňaní hodnôt pritom môže dochádzať k zmenám rozdelenia príznakov, čo je v systéme demonštrované graficky. Systém je zároveň implementovaný tak, aby mohol byť jednoducho rozšírený o ďalšie metódy imputácie a algoritmy strojového učenia. Súčasťou práce je tiež následné experimentálne otestovanie systému na troch zvolených reálnych dátových sadách. Vytvorenie takéhoto systému je inšpirované záverečnou prácou z Masarykovej univerzity [3], ktorá sa venuje zobrazeniu správania algoritmov vzhľadom k charakteru dát.

Výsledok práce je určený študentkám a študentom so záujmom o danú oblasť. Slúži ako edukačný nástroj pre porovnanie zmien vplyvom rôznych metód dopĺňania. Aplikácia zahŕňa sprievodné teoretické texty a ďalej viacero vizualizácií, ktoré sú často nápomocné pri získavaní nových vedomostí.

Nasledujúce časti tejto práce predpokladajú podstatnú znalosť konceptov strojového učenia, ktoré nie sú jej obsahom. Pre lepšie oboznámenie sa s touto problematikou možno odkázať na inú literatúru, akou je napríklad kniha od Murpyho [4].

Kapitola 1

Typy chýbajúcich hodnôt

Pod chýbajúcou hodnotou sa rozumie hodnota, ktorá nie je uvedená pri nejakom z pozorovaných príznakov. Môže byť reprezentovaná niekoľkými formami. Bežne zaužívanými sú hodnota NaN (z angl. *Not a Number*), prázdny reťazec, alebo pomlčka. Na začiatku práce s dátovou sadou je preto nutné identifikovať označenie chýbajúcich hodnôt.

Okrem toho je dôležité venovať pozornosť príčinám ich vzniku. Každá z nich môže totiž inak negatívne ovplyvňovať výsledky analýz, alebo predikcií modelov strojového učenia natrénovaných na takýchto dátach. Podľa príčiny sa rozlišujú celkom tri typy hodnôt: chýbajúce úplne náhodne (MCAR), chýbajúce náhodne (MAR) a nechýbajúce náhodne (MNAR). [5]

Pre presnejšie definovanie pojmov v nasledujúcich podkapitolách zaveďme nasledovné značenie. Majme dátovú sadu $\mathbf{Y}$ s n záznamami a p príznakmi. Označme $\mathbf{R}$ $n \times p$ maticu obsahujúcu 0 a 1 indikujúcu, či je daný bod prítomný v dátovej sade. Ďalej zaveďme $\mathbf{Y}_{\text{mis}}$ obsahujúcu chýbajúce dátové body, teda hodnoty $\mathbf{Y}$ s $\mathbf{R} = 0$. Alternatívne majme $\mathbf{Y}_{\text{obs}}$ obsahujúcu pozorované dátové body, čiže hodnoty $\mathbf{Y}$ s $\mathbf{R} = 1$. Nakoniec označme $\boldsymbol{\psi}$ vektor neznámych parametrov. [6]

1.1 Hodnoty chýbajúce úplne náhodne

Dátové body chýbajú úplne náhodne, ak príčina ich neprítomnosti nie je závislá od dát [7]. Pravdepodobnosť, že budú chýbať nesúvisí s ich hodnotou ani so zvyšnými pozorovanými hodnotami príznakov (viď obr. 1.1) [5]. Matematicky možno zapísať ako

$$P(\mathbf{R} = 0|\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{mis}}, \boldsymbol{\psi}) = P(\mathbf{R} = 0|\boldsymbol{\psi}) \text{ [6]}\,. \tag{1.1}$$

K tomuto typu môže dôjsť v prípade, keď respondent omylom preskočí otázku, alebo keď nastane porucha nástroja zbierajúceho údaje.

Veľkou výhodou hodnôt chýbajúcich úplne náhodne je, že analýzy dátových súborov ostávajú nezaujaté. Odhadované parametre nie sú nijak ovplyvnené absenciou údajov. [5]

1.2 Hodnoty chýbajúce náhodne

Za dátový bod chýbajúci náhodne sa považuje taký bod, ktorého dôvod chýbania súvisí s hodnotou iného príznaku v dátovej sade, avšak nevplýva na to samotná hodnota dátového bodu. Pravdepodobnosť, že bod bude chýbať závisí od súboru pozorovaných hodnôt, ale nezávisí od konkrétnej chýbajúcej hodnoty (viď obr. 1.1). [5] Matematicky

$$P(\mathbf{R} = 0|\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{mis}}, \boldsymbol{\psi}) = P(\mathbf{R} = 0|\mathbf{Y}_{\text{obs}}, \boldsymbol{\psi}) \text{ [6]}\,. \tag{1.2}$$

Príkladom môže byť situácia, kedy cestovná kancelária zbiera odpovede týkajúce sa letnej dovolenky od rôznych respondentov. Destinácia dovolenky chýba najčastejšie v prípade verejne známych osobností. Dôvod, kvôli ktorému sa rozhodli danú otázku nezodpovedať však pravdepodobne nesúvisí so skutočnou destináciou, ale s ochranou ich súkromia, teda s tým, že sú slávni.

Chýbajúce dátové body nie sú rozložené rovnomerne naprieč celou dátovou sadou, ale chýbajú najmä v rámci danej podskupiny [7].

1.3 Hodnoty nechýbajúce náhodne

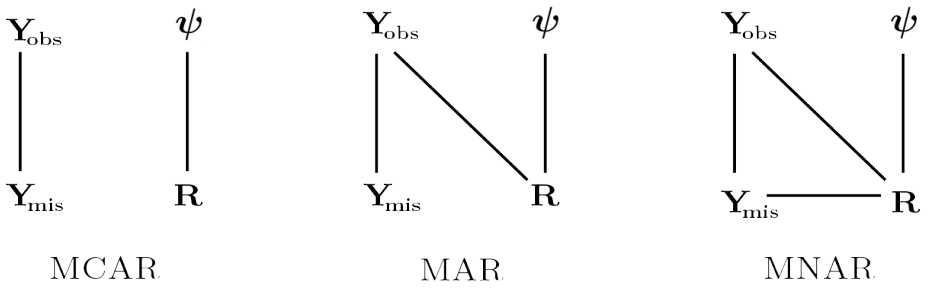
Posledným typom sú dátové body nechýbajúce náhodne. Tento prípad nastáva, keď dôvodom chýbania dátového bodu je jeho samotná hodnota (viď obr. 1.1) [7]. Výraz

$$P(\mathbf{R} = 0|\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{mis}}, \boldsymbol{\psi}) \tag{1.3}$$

teda nemožno nijak zjednodušiť [6].

Jedným z možných prípadov vzniku je situácia, keď respondent nechce odpovedať na otázku kvôli tomu, aká by bola jeho skutočná odpoveď. Napríklad ľudia trpiaci obezitou budú mať pravdepodobne vyššiu tendenciu vynechať otázku týkajúcu sa ich hmotnosti. V takomto prípade by ich hmotnosť nechýbala náhodne, ale dôvodom by bola jej reálna hodnota.

Ide o veľmi problematický typ chýbajúcich hodnôt, keďže hodnoty nie je možné spoľahlivo odhadnúť na základe iných záznamov v súbore [7].



Obr. 1.1 Zobrazenie typov chýbajúcich hodnôt [8]

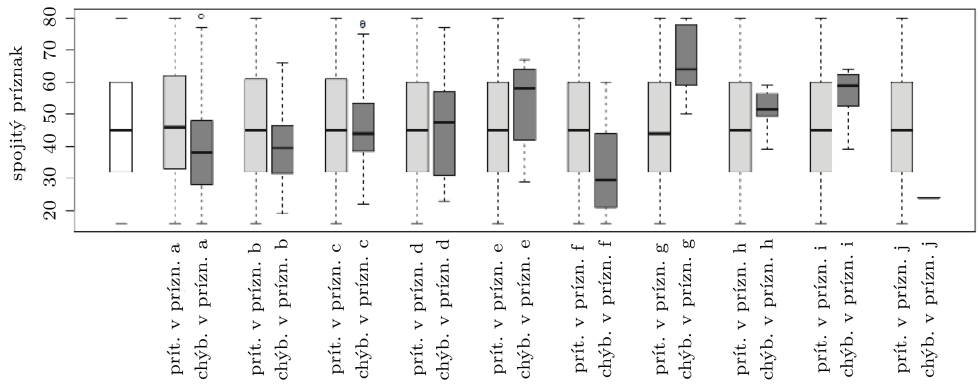
Kapitola 2

Vizualizácia chýbajúcich hodnôt

Vizualizácie nekompletných dátových súborov sú užitočným nástrojom pre skúmanie štruktúry chýbajúcich hodnôt a určenie prípadných vzťahov k dostupným údajom. Môžu tiež pomôcť určiť typ chýbajúcich hodnôt, čo je v praxi pomerne komplikovaná úloha. [9]

2.1 Paralelný krabicový diagram

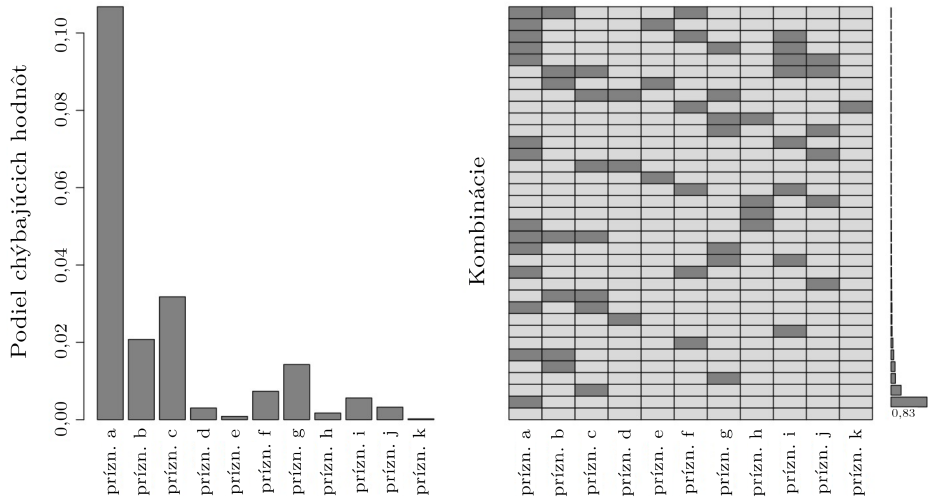
Paralelný krabicový diagram (viď obr. 2.1) sa používa pre vizualizáciu spojitého príznaku. Zobrazuje podmienené rozdelenia podľa jednotlivých príznakov, ktoré majú prekódované hodnoty na chýbajúce a nechýbajúce. Je vhodný najmä pre skúmanie toho, či zobrazený spojitý príznak vysvetľuje rozdelenie chýbajúcich hodnôt v inom príznaku. Na obr. 2.1 je bielou označený štandardný krabicový diagram, svetlosivou sú označené diagramy zoskupené podľa pozorovaných údajov a tmavosivou diagramy zoskupené podľa chýbajúcich údajov. [9]



Obr. 2.1 Paralelný krabicový diagram [9]

2.2 Agregačný diagram

Na obrázku 2.2 sú znázornené agregačné diagramy. Slúžia pre zobrazenie objemu chýbajúcich hodnôt v dátovej sade. Graf v ľavej časti ukazuje podiel chýbajúcich hodnôt v každom príznaku, pričom alternatívne môžu byť využité aj absolútne počty. Graf v pravej časti prezentuje kombinácie, v ktorých sa chýbajúce hodnoty vyskytujú. Na jeho okraji je vykreslený stĺpcový graf, ktorý zobrazuje frekvenciu výskytu jednotlivých kombinácií. Svetlosivá farba reprezentuje pozorované dátové body a tmavosivá tie chýbajúce. [9]

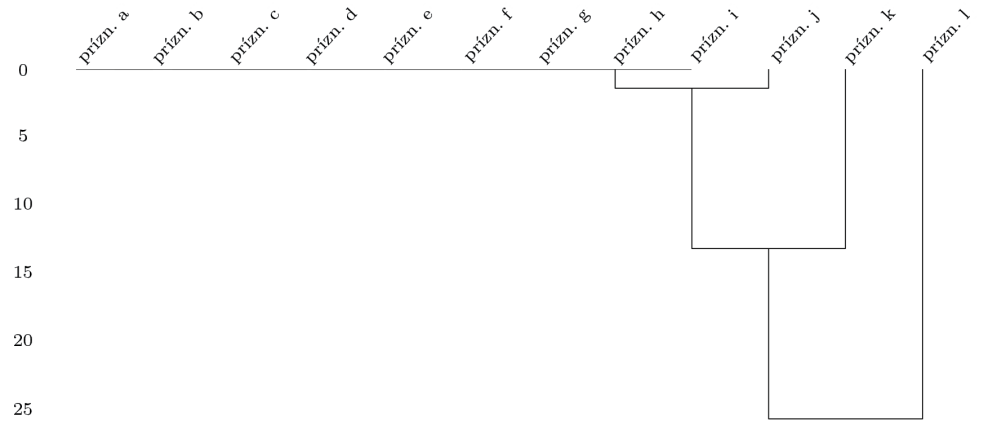


Obr. 2.2 Agregačné diagramy [9]

2.3 Dendrogram chýbajúcich hodnôt

Dendrogram chýbajúcich hodnôt (viď obr. 2.3) je stromový diagram, ktorý využíva algoritmus hierarchického zhlukovania pre zoskupenie jednotlivých príznakov. Funguje na princípe vzájomnej podobnosti, v tomto prípade určenej na základe tzv. korelácie nulity definovanej v rozsahu od -1 po 1. Hodnota -1 znamená, že ak je jeden príznak prítomný, druhý bude určite chýbať. Hodnota 0 znamená, že prítomnosť jedného príznaku nemá žiadny vplyv na absenciu druhého príznaku. Hodnota 1 znamená, že ak je jeden príznak prítomný, druhý bude určite tiež prítomný. [10]

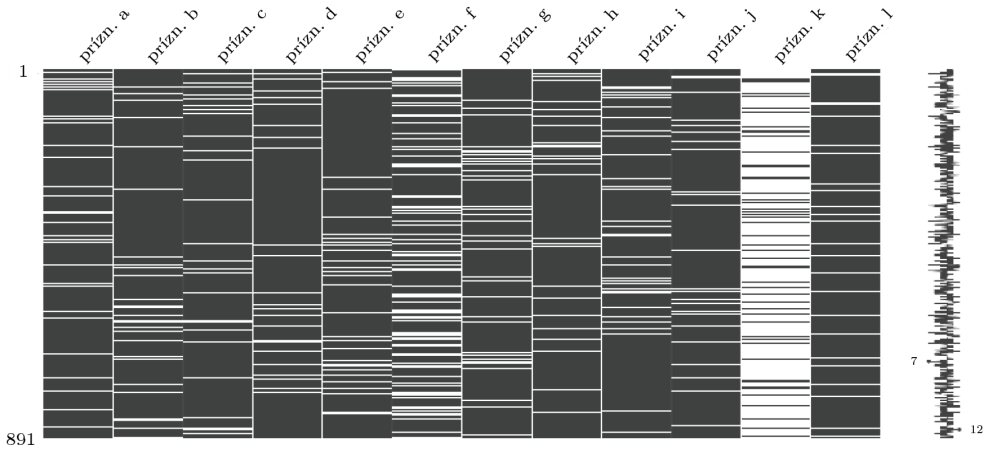
Diagram sa interpretuje pohľadom zhora nadol. Príznaky, ktoré sú spolu prepojené vo vzdialenosti nula si vzájomne plne predpovedajú prítomnosť. Príznaky, ktoré sa delia blízko nuly si navzájom predpovedajú prítomnosť pomerne presne, ale nie dokonale. Výška rozdielu nulity príznakov uvedená na ľavej strane hovorí, koľko hodnôt by bolo nutné doplniť, alebo odstrániť, aby si ich nulita odpovedala. [10]



Obr. 2.3 Dendrogram chýbajúcich hodnôt

2.4 Maticový diagram

Maticový diagram (viď obr. 2.4) zobrazuje všetky body dátového súboru. Jednotlivé hodnoty sú prekódované na chýbajúce a nechýbajúce, na základe čoho sú od seba farebne odlíšené. Diagram je v pravej časti doplnený o čiarový graf, ktorý znázorňuje tvar úplnosti údajov, pričom zvýrazňuje záznamy s najväčšou a najmenšou nulitou. [10] Dáva pohľad na rozdelenie chýbajúcich hodnôt v celom súbore, čo umožňuje zachytiť vzory ich výskytu [9]. Tmavosivá farba označuje pozorované body a biela chýbajúce body.



Obr. 2.4 Maticový diagram

Kapitola 3

Možnosti prístupu k chýbajúcim hodnotám

Od doby, kedy bolo prvýkrát poukázané na možné nebezpečenstvá, ktoré vznikajú v dôsledku ponechania nespracovaných chýbajúcich hodnôt, bolo navrhnutých niekoľko prístupov riešenia. Navzájom sa od seba odlišujú v matematických aj filozofických základoch. [11]

3.1 Odstránenie chýbajúcich hodnôt

Medzi jednoduchšie prístupy patrí odstránenie chýbajúcich hodnôt. Ukazuje sa však, že vedie k nižšej spoľahlivosti dát a všeobecne nie je odporúčané takto postupovať [12]. Obzvlášť v prípade neskoršej implementácie predikčných algoritmov je spravidla požadovaný čo najvyšší počet dostupných údajov. Sú však prípady, kedy môže byť vhodné v rámci jedného dátového súboru niektoré hodnoty doplniť a iné odstrániť.

Existujú dve roviny, z ktorých možno nazerať na objem chýbajúcich hodnôt. Jednou je rovina záznamov, do ktorej spadá analýza kompletných a analýza dostupných prípadov. Druhou je rovina príznakov, v rámci ktorej sa pozoruje vplyv ich redukcie. [13]

3.1.1 Analýza kompletných prípadov

V analýze kompletných prípadov [5] sa odstraňujú všetky záznamy, ktoré majú aspoň jednu hodnotu príznaku chýbajúcu. Môže tak dôjsť k významnému zmenšeniu dátového súboru. Navyše tento prístup zanecháva nezaujaté odhady parametrov len ak sú dáta MCAR.

3.1.2 Analýza dostupných prípadov

V analýze dostupných prípadov [5] dochádza k odstráneniu dátového bodu len vtedy, keď chýba jeho hodnota potrebná pre analýzu, alebo testovanie nejakého predpokladu. Ak sa teda v dátovom súbore nachádza záznam, ktorému chýba hodnota pri nejakom danom príznaku, tento záznam bude môcť byť stále použitý pre analýzu zvyšných príznakov, alebo pre testovanie iných predpokladov.

V tomto prípade sa zachová väčšie množstvo informácie, keďže sa využijú všetky pozorované údaje. Problém takéhoto prístupu však spočíva v tom, že spočítané štatistiky, alebo odhady parametrov výsledného modelu sa zakladajú na odlišných podmnožinách dátového súboru. [5]

3.1.3 Redukcia príznakov

Príznaky v dátových súboroch, ktorých hodnoty chýbajú viac ako v 50 % záznamov môže byť vhodné zo súboru úplne odstrániť. Tento prístup však nie je bez rizika. Dokáže zapríčiniť zníženie schopnosti predikcie a spôsobiť skreslenie, ktoré negatívne ovplyvňuje reprezentatívnosť výsledkov [13]. Je adekvátne takto postupovať najmä v prípadoch, kedy príznak nie je významný pre analýzu, alebo trénovaný model.

3.2 Doplnenie chýbajúcich hodnôt

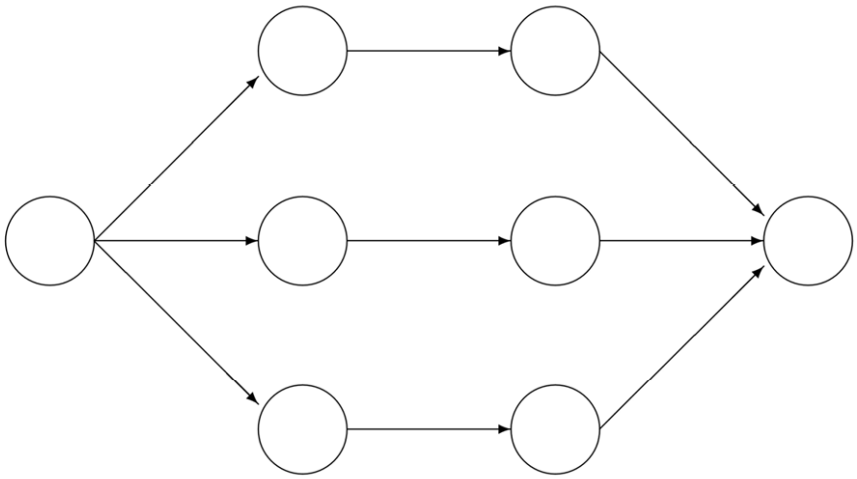
Viacero metód zaobchádzania s chýbajúcimi hodnotami je založených na imputácii. Namiesto odstránenia chýbajúcich dátových bodov sa nahrádzajú vierohodnými odhadmi. Metódy využívajú informácie z dátových súborov a na základe nich sa snažia určiť hodnotu, ktorá je pre daný záznam najpravdepodobnejšia možná za predpokladu jej ostatných príznakov [14]. Výhodou tohto prístupu je, že sa počet pozorovaných údajov nezmenší.

Hodnoty možno doplniť buď na základe štatistických metód, alebo pomocou algoritmov strojového učenia. Výhodou strojového učenia je, že často dokáže zachytiť aj zložitejšie vzťahy medzi príznakmi, čo vedie k presnejším odhadom. Na druhej strane sú jeho výsledky spravidla horšie interpretovateľné v porovnaní so štatistickými metódami. [15]

Okrem toho sa rozdeľujú imputácie na jednoduché a viacnásobné. Jednoduché nájdu jednu spoľahlivú hodnotu, ktorou sa chýbajúci bod nahradí. Nijak pri tom nezohľadňujú neistotu s akou bola táto hodnota zvolená. V niektorých prípadoch však už tento prístup dosahuje lepšie výsledky a poskytuje nestranné odhady. [11]

Viacnásobné imputácie riešia tento problém vykonaním $m > 1$ nezávislých doplnení rovnakého dátového bodu, čím dôjde k vytvoreniu m kompletných dátových súborov. Dátové súbory sa od seba navzájom líšia len na miestach,

kde boli údaje chýbajúce. [11] Rozdiely medzi jednotlivými hodnotami pritom vyjadrujú neistotu s akou boli doplnené [16]. Súbory sa potom analyzujú samostatne a výsledky sa na konci spoja do jedného súhrnného odhadu (viď obr. 3.1). Štandardná chyba kombinuje odchýlky medzi m súbormi s odchýlkami v rámci nich, čo vedie k neskresleným odhadom s dobrými štatistickými vlastnosťami. [6]



Obr. 3.1 Priebeh viacnásobnej imputácie [6]

Kapitola 4

Vplyv doplnenia hodnôt

Po ošetrení chýbajúcich hodnôt vybranou metódou nás zväčša zaujíma, ako úspešná bola a prípadne jej porovnanie s inými metódami. Pri imputácii dát sa presnosť hodnotí buď priamo, alebo nepriamo [17]. Vplyvom doplnenia hodnôt sa venuje niekoľko štúdií, ktoré skúmajú rôzne prípady použitia. Prehľad niektorých z nich je predstavený v nasledujúcich podkapitolách, pričom konkrétne popisy použitých metód možno nájsť v odkazovaných prácach.

4.1 Priame hodnotenie

Priame hodnotenie imputácie prebieha na kompletnej dátovej sade, z ktorej sú umelo odstránené niektoré body. Tie sú následne doplnené a porovnáva sa ich skutočná hodnota s tou doplnenou, pričom sa posudzuje veľkosť odchýlky. [17]

Kategorické a diskrétne príznaky sa často vyhodnocujú pomocou percenta správnych odpovedí. Pre vyhodnotenie spojitých príznakov je zvykom použiť odmocninu strednej kvadratickej chyby (RMSE) definovanú ako

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{x_i})^2}, \tag{4.1}$$

kde n je počet záznamov, x_i sú skutočné a $\widehat{x_i}$ predikované hodnoty. [17]

Priame hodnotenie vo svojej štúdii aplikovali Sun a spol. [18] pre porovnanie viacrozmernej imputácie pomocou zreťazených rovníc (MICE), generatívnych súperiacich imputačných sietí (GAIN), variačného autoenkodéra (VAE) a imputácie pomocou náhodného lesa (MissForest). Použili desať úplných tabulárnych dátových sád, z ktorých sedem bolo reálnych a tri generované s navzájom rôznymi charakteristikami. Dátové body odstraňovali v rôznych scenároch tak, aby dosiahli MCAR, MAR aj MNAR. Skúšali dopĺňať spojité aj kategorické príznaky. Výkon metód porovnávali podľa RMSE a percenta správnych odpovedí. Zistili, že konvenčné metódy MICE aj MissForest prekonali metódy

hlbokého učenia GAIN aj VAE. Metóda GAIN dosahovala dobré výsledky len v prípade, kedy chýbajúce dáta boli typu MCAR. Dong a spol. [19] ďalej podotkli, že metódy hlbokého učenia dosahujú vyššiu presnosť hlavne na výrazne väčších dátových súboroch.

V inej štúdii sa Platias a spol. [20] zamerali na porovnanie metód k-najbližších susedov (kNN), MICE, MissForest, iteratívnej imputácie, autoenkodéra a ďalších menej známych. Experimentovali so štyrmi reálnymi kompletnými dátovými sadami, z ktorých náhodne odstraňovali rôzne objemy bodov. Presnosť určovali na základe strednej absolútnej chyby a RMSE, no metódy s najlepšími výsledkami sa naprieč experimentmi líšili. Nedospeli teda k jednej, ktorá by jednoznačne predčila zvyšné. Ukázalo sa však, že použitie MissForest aj MICE viedlo často k spoľahlivým odhadom. MICE pritom dosahovalo dobré výsledky aj pri veľkom objeme chýbajúcich hodnôt. V prípade, že obe tieto metódy zlyhali, sa kNN ukázalo byť dobrou alternatívou. Naopak autoenkodéry v nimi vykonaných experimentoch dosiahli celkovo horšie výsledky. Iteratívna imputácia dokonca v situáciách, keď chýbala polovica záznamov, miestami ani nebola schopná vrátiť nejaký odhad a fungovala len pri menšom počte chýbajúcich hodnôt.

Autori práce [21] skúmali MissForest, MICE, algoritmus striedania vzoru chýbania Lasso (MissPALasso) a kNN. Využili rôzne kompletné dátové sady, z ktorých niektoré mali všetky príznaky spojité, všetky príznaky kategorické, alebo zmiešané typy príznakov. Náhodne z nich odstraňovali body tak, aby ich bolo postupne celkovo chýbajúcich 10 %, 20 % a 30 %. Pri čisto spojitých príznakoch boli využité štyri dátové sady a presnosť merali na základe normalizovanej RMSE. MissForest dosahovala lepšie výsledky v porovnaní s kNN. MissPALasso mal v niektorých prípadoch o trochu menšiu chybu ako MissForest, avšak úplne zlyhal pri jednej dátovej sade, kde mu vysoká dimenzionalita znemožnila výpočet. Dátové sady obsahujúce len kategorické príznaky boli tri a presnosť hodnotili na základe percenta správnych odpovedí. MissForest mala najlepšie výsledky spomedzi MICE aj kNN. Ďalšie experimenty vykonali na štyroch dátových sadách so spojitými aj kategorickými príznakmi. Tiež sa ukázalo, že MissForest fungovala najspoľahlivejšie z porovnávaných metód.

4.2 Nepriame hodnotenie

Nepriame hodnotenie využíva na posúdenie kvality modely strojového učenia, ktoré sú natrénované na sade s doplnenými hodnotami. Výkon takéhoto modelu je následne posudzovaný prostredníctvom evaluačných metrík. Hlavná myšlienka za tým je, že model s lepšími výsledkami za použitia rovnakej metriky indikuje vyššiu kvalitu imputácie, v dôsledku čoho dosahuje lepšie predikcie. [17]

Volené modely strojového učenia sú supervizované, ktoré riešia klasifikačné a regresné problémy. Podľa typu modelu sa volí vhodná evaluačná metrika. Pre klasifikátory sa zvykne počítať percento správnych predikcií, alebo plocha

pod krivkou operačnej charakteristiky prijímača (AUC), kým pre regresory sa používajú stratové funkcie ako RMSE.

Nepriame hodnotenie využila Lynette A. Hunt [22] pre porovnanie metód kNN, MissForest, imputácie priemerom, mediánom a tzv. hot-deck imputácie. Použila štyri dátové sady, z ktorých ponechala len kompletné záznamy. Následne z nich odstránila 10 %, 20 %, 30 % a 50 % hodnôt. Na takto vytvorených dátových sadách natrénovala niekoľko modelov strojového učenia, ako napr. rozhodovací strom, naivný Bayes, logistickú regresiu, či kNN. Skúmala aký vplyv má percento chýbajúcich hodnôt a použitá metóda imputácie na presnosť predikcií. Z jej výsledkov vyplynulo, že pre použité dátové sady imputácia priemerom, mediánom a kNN dosahovala približne rovnaké výsledky čo sa presnosti predikcií týka. Navyše v týchto prípadoch platilo, že čím viac chýbajúcich hodnôt sa vyskytovalo v súbore, tým bola presnosť výsledného modelu menšia. Hot-deck imputácia a MissForest mali priemerne vyššiu presnosť v porovnaní s ostatnými. Modely natrénované na nimi doplnených dátových súboroch dosahovali približne rovnakú presnosť nezávisle na počte chýbajúcich hodnôt.

Yoon a spol. [23] vykonali experimenty na piatich kompletných dátových sadách. Náhodne z nich odstránili 20 % hodnôt. Tieto body následne doplnili pomocou algoritmov GAIN, MICE, MissForest, doplnenia matice, autoenkodéra a maximálneho vierohodného odhadu. Na takto získaných nových sadách natrénovali model logistickej regresie. Experiment vždy desaťkrát zopakovali a presnosť reportovali pomocou priemerného AUC. Vo všetkých prípadoch dosiahol najlepšie výsledky algoritmus GAIN. V závislosti od použitej sady dosahovali podobné výsledky aj metódy MICE, MissForest a autoenkodér. O niečo horšie dopadlo doplnenie matice a maximálny vierohodný odhad. Okrem toho v tejto štúdii na jednej zvolenej dátovej sade odstraňovali väčšie objemy bodov od 10 % až po 90 %, ktoré následne odhadovali použitím GAIN, MissForest a autoenkodéra. Pri väčších počtoch chýbajúcich bodov už presnosť predikcie viac závisela na doplnení správnych hodnôt. Z experimentov vyplynulo, že presnosť pri použití GAIN bola výrazne vyššia oproti zvyšným metódam.

Jerez a spol. [15] predikovali návrat nádoru onkologických pacientov pomocou neurónových sietí. Dátovú sadu s chýbajúcimi hodnotami upravili analýzou kompletných prípadov a natrénovali na nej referenčný model. Ten porovnávali na základe AUC s modelmi, pri ktorých použili rôzne metódy dopĺňania hodnôt. Experimentovali s imputáciou priemerom, hot-deck imputáciou, MICE, Ameliou, kNN, či viacvrstvovým perceptrónom. Vo všetkých prípadoch s výnimkou hot-deck imputácie bola presnosť vyššia po doplnení hodnôt, ako po ich odstránení. Najvyššia hodnota AUC bola dosiahnutá pri použití kNN.

Okrem jednotlivých chýbajúcich hodnôt rozmiestnených v rámci príznakov však môžu absentovať aj celé príznaky. Touto problematikou sa zaoberá štúdia [24], ktorá skúma ich dopĺňanie pomocou VAE s ľubovoľným podmienením, GAIN, Wassersteinovej modifikácie GAIN (WGAIN), autoenkodéra odstraňujúceho šum, kNN a MICE. Experimenty boli vykonané na reálnych aj umelo

vytvorených dátových sadách so spojitými príznakmi, ktorých objem chýbania celých príznakov sa naprieč experimentmi líšil od 10 % po 50 %. Autori v rámci nepriameho hodnotenia využili viacero klasifikátorov, a to logistickú regresiu, viacvrstvový perceptrón, kNN, stromy s extrémnym zosilnením gradientu, náhodné lesy a naivného Bayesa. Klasifikačné modely natrénovali na kompletných súboroch a pozorovali zmenu ich presnosti na nových dátach s doplnenými chýbajúcimi príznakmi. Z výsledkov experimentov vyplynulo, že GAIN aj WGAIN boli najlepšími metódami imputácie v porovnaní s ostatnými skúšanými. Keď sa objem chýbajúcich príznakov pohyboval do 30 %, dosahoval najlepšie výsledky WGAIN, inak GAIN.

Kapitola 5
Zvolené metódy

Na základe výsledkov štúdií sa ukazuje, že druh použitej metódy pre doplnenie hodnôt v dátovej sade má dopad na presnosť predikcií modelov strojového učenia. Podľa štúdie [17] patrí medzi často využívané metódy v literatúre z rokov 2010–2021 maximálny vierohodný odhad, hot-deck imputácia, imputácia mediánu, MICE, k-means, klasifikačné a regresné stromy, kNN a MissForest.

Z priameho aj nepriameho hodnotenia niekoľkých experimentov vyplýva, že pre konkrétnu problematiku je vhodné skúsiť rôzne metódy a na základe dosiahnutých výsledkov sa rozhodnúť, ktorú použiť. MICE, kNN a MissForest boli v štúdiách často na prvých miestach v porovnaní s inými čo sa presnosti imputácií týka. Okrem toho dosahovali aj výraznejšie zlepšenie schopnosti predikcie. Metóda GAIN sa podľa záverov [23, 24] tiež zdá byť dobrou alternatívou. Veľmi známou metódou je imputácia priemerom, ktorá má hlavne pri zvážení jej nízkej náročnosti implementácie mnohokrát dostačujúce výsledky. Na základe toho bolo do aplikácie vybratých celkom päť metód, a to kNN, imputácia priemerom, MICE, MissForest a GAIN. Voľba týchto metód zároveň pokrýva viaceré typy imputácií, či už podľa princípu, na ktorom zakladajú, alebo podľa počtu vykonaných doplnení.

5.1 Imputácia pomocou k-najbližších susedov

K-najbližších susedov je algoritmus strojového učenia založený na podobnosti, čo sa využíva práve pri imputácii. Chýbajúca hodnota môže byť odhadnutá na základe hodnôt iných bodov, ktoré sú k nej najbližšie [25].

Algoritmus možno popísať nasledujúcimi krokmi:

1. Zvolenie chýbajúcej hodnoty z ľubovoľného záznamu, ktorá bude dopĺňaná.
2. Spočítanie vzdialenosti medzi zvyšnými príznakmi zvoleného záznamu s príznakmi ostatných záznamov v dátovej sade, ktoré majú dopĺňaný príznak prítomný. Vzdialenosť je počítaná len na základe príznakov, ktoré sú prítomné v oboch záznamoch.

3. Výber k záznamov s najnižšou vzdialenosťou.

4. Spočítanie váženého priemeru k hodnôt dopĺňaného príznaku. Takto získaný odhad sa použije pre doplnenie. [20]

Vzdialenosť je väčšinou počítaná pomocou Euklidovskej vzdialenosti definovanej za prítomnosti chýbajúcich hodnôt ako

$$D_{ij} = \sqrt{\frac{d}{p}\sum_{k=1}^{p}(\mathbf{X}_{ik} - \mathbf{X}_{jk})^2}\,, \tag{5.1}$$

kde D_{ij} označuje vzdialenosť medzi i-tým a j-tým záznamom matice dát $\mathbf{X}$ s rozmermi $n \times d$ a p označuje počet príznakov, ktoré sú prítomné v oboch záznamoch súčasne [26]. Keďže sa jedná o metódu založenú na počítaní vzdialenosti je dobré príznaky pred jej aplikovaním normalizovať.

V aplikácii je táto metóda implementovaná s využitím triedy `KNNImputer`[1] z knižnice `scikit-learn`. Pred jej použitím sú nenumerické príznaky zakódované využitím tzv. fiktívnych príznakov (z angl. *dummy variables*) a všetky príznaky sú normalizované pomocou min-max normalizácie. Potom nasleduje samotná imputácia, pričom má užívateľ v aplikácii možnosť zvoliť počet susedov k. Vzdialenosť sa vyhodnocuje na základe Euklidovskej vzdialenosti 5.1 a všetkým k susedom je priradená rovnaká váha.

Chýbajúce hodnoty nenumerických príznakov sa na konci nahradia tou kategóriou, ktorej odpovedá fiktívny príznak s najvyššou doplnenou hodnotou v danom zázname, teda vlastne najčastejšou kategóriou medzi k susedmi. Chýbajúce hodnoty numerických príznakov označené ako kategorické sa zaokrúhlia k najbližšej kategórii a chýbajúce hodnoty celočíselných príznakov sa zaokrúhlia k najbližšiemu celému číslu.

5.2 Imputácia priemerom

Ďalšou z metód je imputácia priemerom. Bolo navrhnutých niekoľko variant, pričom obvykle sa používa aritmetický priemer. Ten sa spočíta pre každý príznak z dostupných hodnôt a nahradia sa ním chýbajúce hodnoty v odpovedajúcich príznakoch [27]. Jedná sa v podstate o extrémny prípad metódy kNN, kde za k sú zvolení všetci susedia a priemer nie je vážený vzdialenosťou [13].

V aplikácii sú chýbajúce nekategorické príznaky nahrádzané aritmetickým priemerom a kategorické príznaky modusom. Ak je navyše príznak celočíselný, zaokrúhli sa k najbližšiemu celému číslu.

Pri dopĺňaní hodnôt touto metódou sa neberú do úvahy vzťahy medzi príznakmi. Týmto prístupom sa zmenší rozptyl príznaku ako aj jeho kovariancia s ostatnými príznakmi. Metóda zanecháva skreslené odhady bez ohľadu na typ

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html`

chýbajúcich hodnôt. [27, 13] Veľkou prednosťou však je jednoduchá implementácia a nízka výpočtová náročnosť.

5.3 Viacrozmerná imputácia pomocou zreťazených rovníc

Viacrozmerná imputácia pomocou zreťazených rovníc je viacnásobnou metódou dopĺňania chýbajúcich hodnôt. Metóda predpokladá, že dáta, na ktorých je použitá chýbajú náhodne. Ak nie je tento predpoklad splnený, metódu možno stále použiť, avšak s rizikom, že získané odhady budú viac skreslené. [28]

Algoritmus MICE [29] získava odhady pomocou reťazca podmienených regresných modelov, ktoré využívajú informácie z ostatných príznakov v dátovom súbore. Každý príznak obsahujúci chýbajúce hodnoty je odhadovaný pomocou modelu, ktorý berie na vstupe množinu prediktorov a na výstupe vracia doplnené hodnoty v cieľovom príznaku.

Metódu možno opísať pomocou nasledujúcich krokov:

1. Prvotné doplnenie všetkých chýbajúcich hodnôt v dátovom súbore jednoduchou imputačnou metódou, akou je napríklad doplnenie priemeru, alebo mediánu.
2. Opätovné označenie doplnených hodnôt zvoleného príznaku za chýbajúce.
3. Vytvorenie a natrénovanie regresného modelu, v ktorom je vysvetľovanou premennou zvolený príznak a vysvetľujúcimi premennými je ľubovoľná neprázdna podmnožina zvyšných príznakov súboru.
4. Nahradenie chýbajúcich hodnôt zvoleného príznaku predikciami regresného modelu.

Druhý až štvrtý krok sú následne zopakované pre každý príznak dátovej sady. Zopakovanie týchto krokov pre každý príznak dátového súboru s chýbajúcou hodnotou sa nazýva iteráciou. Príznaky možno prechádzať v rôznom poradí. Zvyčajne sa dopĺňajú zľava doprava, ale možno ich voliť tiež podľa počtu chýbajúcich hodnôt. Na konci každej iterácie sú všetky chýbajúce hodnoty nahradené predikciami, ktoré odrážajú vzájomné vzťahy medzi príznakmi. [29]

Iterácie sa niekoľkokrát vykonajú a počas každej sú aktualizované odhady hodnôt. Na ich konci je výsledkom jeden kompletný dátový súbor. Po dokončení iterácií je celý proces imputácie zopakovaný znovu, čím sa získa ďalšia kompletná dátová sada. [29]

Nevýhodou tejto metódy je, že pre každý dopĺňaný príznak musí byť natrénovaný nový model, čo môže byť pomerne výpočtovo náročné [1].

Vo vytvorenej aplikácii je táto metóda implementovaná len pre numerické príznaky. Prípadné nenumerické sa pred jej použitím najskôr doplnia modusom a zakódujú pomocou fiktívnych premenných.

Pre doplnenie numerických príznakov je použitá trieda `IterativeImputer`[2] z knižnice `scikit-learn`. Prvotná imputácia sa vykonáva pomocou doplnenia priemeru a ako regresný model je použitá Bayesova hrebeňová regresia. Kompletné dátové sady sú na konci združené do jednej priemerovaním doplnených hodnôt, čo je v súlade so štúdiou [30]. Numerické príznaky označené ako kategorické sa zaokrúhlia k najbližšej kategórii a celočíselné príznaky k najbližšiemu celému číslu. V rozhraní aplikácie má užívateľ možnosť zvoliť počet iterácií, ako aj počet vytvorených kompletných dátových súborov.

5.4 Imputácia pomocou náhodného lesa

Imputácia pomocou náhodného lesa [18] je iteratívnou metódou doplňania hodnôt. Vstupom je $n \times d$ matica dátového súboru $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_d)$. Jej ľubovoľný príznak $\mathbf{X}_s$ obsahuje chýbajúce hodnoty na miestach $\mathbf{i}_{mis}^{(s)} \subseteq \{1, 2, ..., n\}$. Rozdeľme podľa neho $\mathbf{X}$ na štyri časti:

- $\mathbf{y}_{obs}^{(s)}$ nech sú hodnoty $\mathbf{X}_s$, ktoré nechýbajú,
- $\mathbf{y}_{mis}^{(s)}$ nech sú hodnoty $\mathbf{X}_s$, ktoré chýbajú,
- $\mathbf{x}_{obs}^{(s)}$ nech sú všetky ostatné príznaky so záznamami $\{1, 2, ..., n\} \setminus \mathbf{i}_{mis}^{(s)}$,
- $\mathbf{x}_{mis}^{(s)}$ nech sú všetky ostatné príznaky so záznamami $\mathbf{i}_{mis}^{(s)}$. [21]

Priebeh algoritmu potom možno popísať ako:

1. Prvotné doplnenie všetkých chýbajúcich hodnôt v dátovom súbore jednoduchou imputačnou metódou, akou je napríklad doplnenie priemeru, alebo mediánu.
2. Natrénovanie náhodného lesa pre predikciu $\mathbf{y}_{obs}^{(s)}$ na základe $\mathbf{x}_{obs}^{(s)}$.
3. Využitie náhodného lesa pre predikciu $\mathbf{y}_{mis}^{(s)}$ na základe $\mathbf{x}_{mis}^{(s)}$.

Druhý a tretí krok sa vykoná pre každý príznak s chýbajúcimi hodnotami, čo predstavuje jednu iteráciu. Tieto kroky sa opakujú, až kým sa rozdiel medzi novo získanou maticou dát a predchádzajúcou maticou dát prvýkrát nezvýši, alebo kým nie je dosiahnutý maximálny počet iterácií. [21]

V aplikácii je táto metóda implementovaná pre numerické aj nenumerické príznaky. Do veľkej miery zakladá na triede `MissForest`[3]. Niektoré časti však boli upravené. Pôvodná implementácia využíva znalosti z testovacej množiny pri trénovaní, čo nebolo žiadúce. Z tohto dôvodu boli odpovedajúce kúsky implementácie zmenené. Ďalej bola nahradená pôvodne používaná knižnica

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html`

[3] `https://github.com/yuenshingyan/MissForest`

`LightGBM` za knižnicu `scikit-learn`. Okrem toho bolo upravené zastavovacie kritérium tak, aby odpovedalo štúdii [21].

Pred doplnením hodnôt sú nenumerické príznaky zakódované tak, že každej kategórii odpovedá jedno číslo. Podľa záverov [31] sa jedná o vhodnú alternatívu ku kódovaniu pomocou fiktívnych príznakoch v prípadoch, kedy je kardinalita kategórií vysoká, pretože dosahuje porovnateľnú presnosť pri nižšej časovej náročnosti trénovania modelu.

Potom nasleduje samotná imputácia, v rámci ktorej sa pre odhad chýbajúcich hodnôt kategorických príznakov používa `RandomForestClassifier` a pre odhad chýbajúcich hodnôt nekategorických príznakov `RandomForestRegressor` z knižnice `scikit-learn`.

Zastavovacie kritérium sa vyhodnocuje na základe rozdielu medzi doplnenými maticami. Rozdiel sa počíta osobitne pre kategorické a nekategorické príznaky. Kritérium sa splní, ak rozdiel prvýkrát narastie pre aspoň jeden z typov. Rozdiel nekategorických príznakov $\mathbf{N}$ sa počíta ako

$$NekatRoz = \frac{\sum_{j \in \mathbf{N}} (\mathbf{X}^{imp}_{new} - \mathbf{X}^{imp}_{old})^2}{\sum_{j \in \mathbf{N}} (\mathbf{X}^{imp}_{new})} \tag{5.2}$$

a rozdiel kategorických príznakov $\mathbf{F}$ ako

$$KatRoz = \frac{\sum_{j \in \mathbf{F}} \sum_{i=1}^{n} \mathbf{I}_{\mathbf{X}^{imp}_{new} \neq \mathbf{X}^{imp}_{old}}}{\#NaN}, \tag{5.3}$$

kde $\#NaN$ označuje počet chýbajúcich hodnôt v kategorických príznakoch. [21]

Celočíselné príznaky sa na konci po doplnení ešte zaokrúhlia k najbližšiemu celému číslu.

5.5 Generatívne súperiace imputačné siete

Majme $\mathbf{X} = (X_1, ..., X_d)$ náhodnú veličinu nazývanú dátový vektor a majme $\mathbf{M} = (M_1, ..., M_d)$ z $\{0, 1\}^d$ náhodnú veličinu nazývanú vektor masky, ktorá označuje prítomné časti dátového vektora. Ďalej označme $\widetilde{\mathbf{X}} = (\tilde{X}_1, ..., \tilde{X}_d)$ ako

$$\tilde{X}_i = \begin{cases} X_i, & \text{ak } M_i = 1 \\ *, & \text{inak} \end{cases}, \tag{5.4}$$

kde $*$ predstavuje chýbajúce hodnoty. [23]

Generatívne súperiace imputačné siete sa potom používajú pre doplnenie týchto hodnôt a predstavujú zástupcu modelu hlbokého učenia. Skladajú sa z dvoch komponent: generátora G a diskriminátora D (viď obr. 5.1) [18].

Generátor dostáva na vstupe $\widetilde{\mathbf{X}}$, $\mathbf{M}$ a vektor šumu $\mathbf{R} = (R_1, ..., R_d)$ nezávislý od $\widetilde{\mathbf{X}}$ a $\mathbf{M}$. Jeho úlohou je doplniť chýbajúce hodnoty $\widetilde{\mathbf{X}}$, pričom sa

snaží čo najviac zmiasť diskriminátor. Výstupom generátora je vektor $\hat{\mathbf{X}}$ s už odhadnutými hodnotami. Matematicky

$$\begin{aligned} \bar{\mathbf{X}} &= G(\widetilde{\mathbf{X}},\ \mathbf{M},\ (\mathbf{1}-\mathbf{M})\odot\mathbf{R})\,, \\ \hat{\mathbf{X}} &= \mathbf{M}\odot\widetilde{\mathbf{X}} + (\mathbf{1}-\mathbf{M})\odot\bar{\mathbf{X}}\,, \end{aligned} \tag{5.5}$$

kde $\odot$ označuje násobenie po zložkách a $\bar{\mathbf{X}}$ vektor po imputácii. [18, 32]

Majme $\mathbf{B} = (B_1, ..., B_d) \in \{0,1\}^d$ náhodnú veličinu definovanú tak, že najskôr sa z $\{0, ..., d\}$ rovnomerne náhodne vyberie k a $\mathbf{B}$ je potom dané ako

$$B_j = \begin{cases} 1, & \text{ak } j \neq k \\ 0, & \text{ak } j = k \end{cases}. \tag{5.6}$$

Zaveďme ďalej d-dimenzionálny vektor nápovied $\mathbf{H}$ získaný náhodným výberom určitých častí z $\mathbf{M}$ ako

$$\mathbf{H} = \mathbf{B}\odot\mathbf{M} + 0{,}5(\mathbf{1}-\mathbf{B}). \text{ [23]} \tag{5.7}$$

Vstup diskriminátora potom tvorí $\hat{\mathbf{X}}$ a $\mathbf{H}$. Diskriminátor sa snaží určiť, ktoré zložky $\hat{\mathbf{X}}$ boli doplnené generátorom. $\mathbf{H}$ pritom upozorňuje na niektoré oblasti $\hat{\mathbf{X}}$, vďaka čomu sa na ne diskriminátor zameria. Výstupom je predikovaný vektor masky $\hat{\mathbf{M}}$ definovaný ako

$$\hat{\mathbf{M}} = D(\hat{\mathbf{X}}, \mathbf{H}). \text{ [18, 32]} \tag{5.8}$$

Diskriminátor je trénovaný tak, aby robil čo najmenej chýb a generátor je naopak trénovaný tak, aby diskriminátor robil chýb čo najviac. Definujme veličinu $V(D,G)$ ako

$$\begin{aligned} V(D,G) = \mathbb{E}_{\hat{\mathbf{X}},\mathbf{M},\mathbf{H}}[&\mathbf{M}^T \log D(\hat{\mathbf{X}},\mathbf{H}) + \\ &(\mathbf{1}-\mathbf{M})^T \log(\mathbf{1}-D(\hat{\mathbf{X}},\mathbf{H}))]\,, \end{aligned} \tag{5.9}$$

kde log predstavuje logaritmus po zložkách a závislosť na G je skrz $\hat{\mathbf{X}}$. Pri trénovaní je tak cieľom

$$\min_{\mathrm{G}}\ \max_{\mathrm{D}}\ V(D,G)\,. \text{ [23]} \tag{5.10}$$

V aplikácii je táto metóda implementovaná len pre numerické príznaky. Prípadné nenumerické sa pred jej použitím najskôr doplnia modusom a zakódujú pomocou fiktívnych premenných.

Samotný GAIN je do veľkej miery prevzatý zo štúdie [23]. Pôvodná implementácia však nepredpokladá použitie trénovacej a testovacej sady, a tak boli niektoré jej časti upravené. Numerické príznaky sú najskôr normalizované pomocou min-max normalizácie. Pre vytvorenie generátora aj diskriminátora je využitá knižnica `tensorflow`. Generátor aj diskriminátor sa skladajú z celkom troch vrstiev. Veľkosť ich vstupu odpovedá dvojnásobku počtu príznakov

dátovej sady. Počet neurónov jednotlivých vrstiev odpovedá počtu príznakov sady. Váhy sú nastavené pomocou Xavierovej inicializácie a algoritmus Adam je zvolený ako optimalizačná metóda pre aktualizáciu váh. Skryté vrstvy využívajú ako aktivačnú funkciu rektifikovanú lineárnu jednotku (z angl. *rectified linear unit*) a výstupná vrstva aktivačnú funkciu sigmoida.

Učenie prebieha pomocou menších dávok $(\tilde{\mathbf{x}}(j), \mathbf{m}(j))$ veľkosti k_D. Pre každú vzorku dávky sa určí $\mathbf{r}(j)$ a $\mathbf{b}(j)$ z $\mathbf{R}$ a $\mathbf{B}$, na základe čoho sa spočíta $\hat{\mathbf{x}}(j)$ a $\mathbf{h}(j)$. Definujme stratovú funkciu $\mathcal{L}_D : \{0,1\}^d \times [0,1]^d \times \{0,1\}^d \to \mathbb{R}$ ako

$$\mathcal{L}_D(\mathbf{m}, \hat{\mathbf{m}}, \mathbf{b}) = \sum_{i:b_i=0} \left[m_i \log\left(\hat{m_i}\right) + (1 - m_i) \log\left(1 - \hat{m_i}\right) \right]. \tag{5.11}$$

Diskriminátor je potom trénovaný tak, aby sa minimalizoval výraz

$$-\sum_{j=1}^{k_D} \mathcal{L}_D(\mathbf{m}(j), \hat{\mathbf{m}}(j), \mathbf{b}(j)). \tag{5.12}$$

Zaveďme ďalej ďalšie dve stratové funkcie $\mathcal{L}_G : \{0,1\}^d \times [0,1]^d \times \{0,1\}^d \to \mathbb{R}$ ako

$$\mathcal{L}_G(\mathbf{m}, \hat{\mathbf{m}}, \mathbf{b}) = -\sum_{i:b_i=0} (1 - m_i) \log\left(\hat{m_i}\right) \tag{5.13}$$

a $\mathcal{L}_M : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ ako

$$\mathcal{L}_M(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} m_i (x_i' - x_i)^2. \tag{5.14}$$

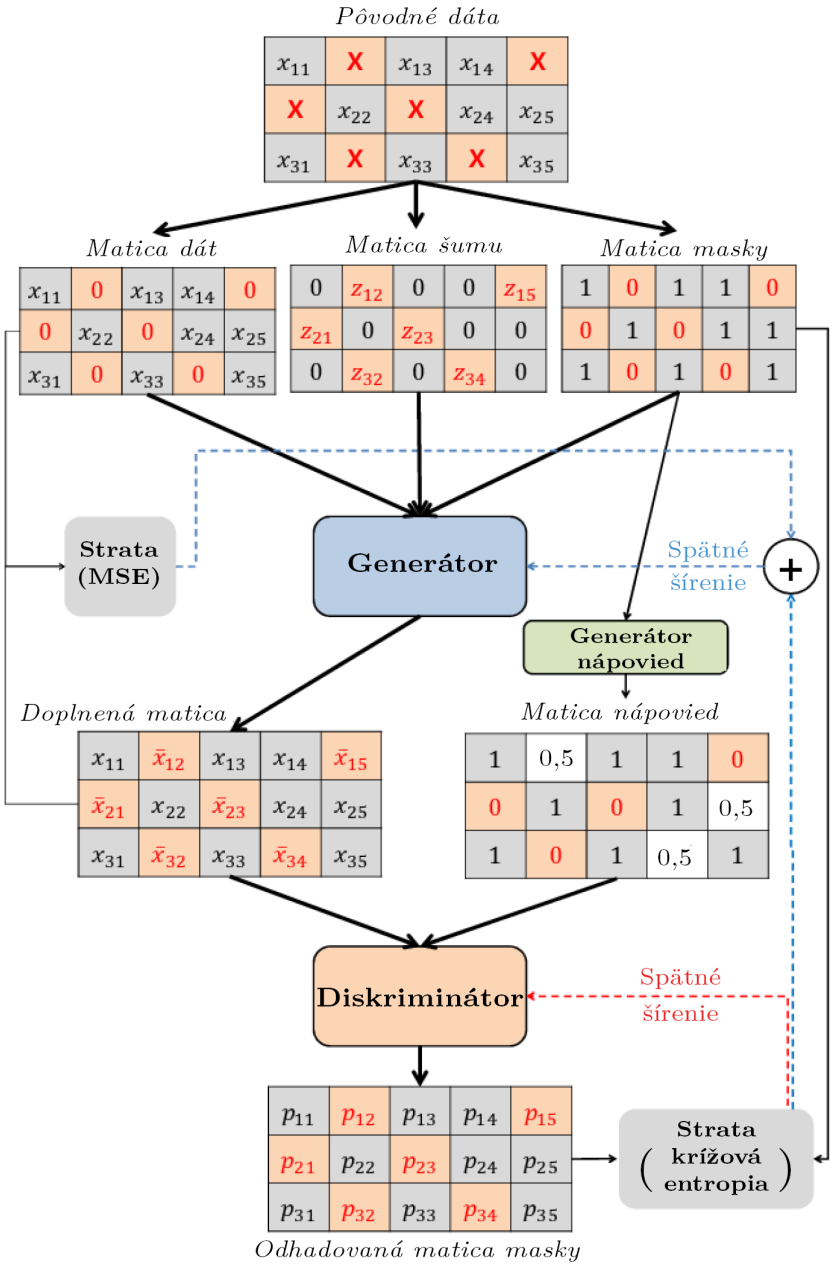
Generátor je trénovaný tak, aby sa minimalizoval výraz

$$\sum_{j=1}^{k_D} \mathcal{L}_G(\mathbf{m}(j), \hat{\mathbf{m}}(j), \mathbf{b}(j)) + \alpha \mathcal{L}_M(\tilde{\mathbf{x}}(j), \hat{\mathbf{x}}(j)), \tag{5.15}$$

kde α je hyperparameter definujúci váhu, ktorá má byť pri učení kladená na presnosť predikcií. [23]

Užívateľ má v rámci aplikácie možnosť zvoliť hyperparameter α, veľkosť dávky k_D, mieru nápovied a počet iterácií vykonaný pri trénovaní generátora a diskriminátora. Zvolená miera nápovied hovorí, s akou pravdepodobnosťou budú jednotlivé hodnoty matice masky prezradené diskriminátoru. Vykonaný počet iterácií však môže byť aj menší ako zvolil užívateľ v prípade, že sa objavia hodnoty NaN či INF (z angl. *infinite*). Vyskytujú sa najmä v situáciách, keď vstupy neurónových sietí spôsobia neplatné hodnoty parametrov, v dôsledku čoho sa trénovanie ukončí predčasne [33].

Po dobehnutí algoritmu GAIN sa numerické príznaky označené ako kategorické zaokrúhlia k najbližšej kategórii a celočíselné numerické príznaky sa zaokrúhlia k najbližšiemu celému číslu.



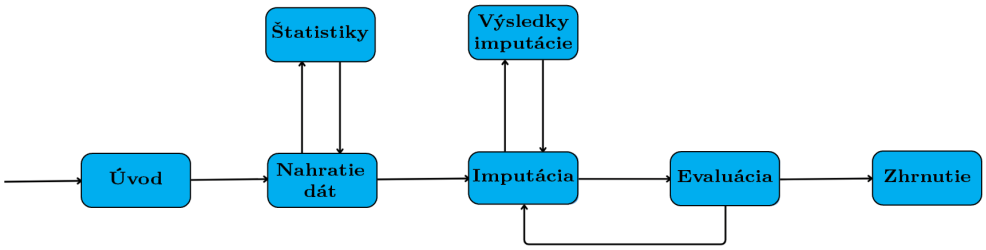
Obr. 5.1 Priebeh algoritmu GAIN [23]

Kapitola 6

Implementovaný systém

Vytvorená webová aplikácia je implementovaná v jazyku `Python` s použitím knižnice `Dash`[1], ktorá bola zvolená najmä kvôli úzkemu prepojeniu s grafickou knižnicou `Plotly`[2]. Skladá sa celkom zo siedmych stránok, ktoré sú znázornené na obr. 6.1. Jednotlivé stránky obsahujú v úvode kratší text s pokynmi pre používanie.

V tejto aplikácii má užívateľ možnosť nahrať dátovú sadu, doplniť jej chýbajúce hodnoty rôznymi metódami a na takto doplnených sadách natrénovať viaceré modely strojového učenia. Úspešnosť jednotlivých modelov v závislosti na imputačnej metóde je na konci zobrazená graficky. Porovnávaný je tiež čas trvania jednotlivých metód.



Obr. 6.1 Schéma pohybu medzi stránkami aplikácie

6.1 Práca s dátovou sadou

Po spustení systému sa ako prvá otvorí úvodná stránka zobrazená na obr. A.1 v prílohe, ktorá predstavuje aplikáciu.

Ďalej nasleduje stránka, ktorá umožňuje užívateľovi nahrať ľubovoľný súbor s dátami vo formáte CSV s veľkosťou do 100 MB. Veľkostné obmedzenie je dané

[1] `https://dash.plotly.com/`
[2] `https://plotly.com/python/`

používanou komponentou `dcc.Upload` z knižnice `Dash`, no súvisí tiež s časovou náročnosťou používaných metód imputácie.

Po nahratí dátovej sady sa zobrazia jej hodnoty formou tabuľky s možnosťou filtrovať ich podľa čísla, reťazca, alebo chýbajúcich hodnôt. Okrem toho možno zvoliť dátové typy jednotlivých príznakov, pričom je na výber z typov `float` (číslo s plávajúcou desatinnou čiarkou), `int` (celé číslo) a `category` (kategória). Dátové typy ovplyvňujú ako budú príznaky vizualizované a spôsob, akým budú dopĺňané. Okrem toho je možné jednotlivé príznaky zo sady odstraňovať. Ukážka stránky je zobrazená na obr. A.2 v prílohe.

V tejto fáze tiež dochádza k rozdeleniu nahratej sady na trénovaciu a testovaciu množinu, kde trénovacia množina tvorí 80 % pôvodnej dátovej sady. Validačná množina nie je použitá. V aplikácii nedochádza k ladeniu hyperparametrov takým spôsobom, že by sa na základe toho vybrala finálna metóda a finálny model. Skôr je každý model v kombinácii s každou skúšanou metódou považovaný za finálny a tieto výsledky sú na konci porovnané.

Aby bolo možné natrénovať modely strojového učenia, musí byť zvolený príznak, ktorý bude predikovaný a typ problému (klasifikačný alebo regresný). Ak zvolený príznak obsahuje chýbajúce hodnoty, sú príslušné záznamy odstránené. Problematike doplnenia chýbajúcich hodnôt v cieľovom príznaku sa venuje tzv. imputácia označenia (z angl. *label imputation*) [34], ktorá však nie je súčasťou tejto práce.

Okrem toho sú z dátovej sady odstránené príznaky, ktorých objem chýbajúcich bodov je väčší ako 80 %. Doplnenie takýchto príznakov predstavuje náročnejší problém, keďže je k dispozícii len málo informácií o ich hodnotách. Rovnako je vďaka tomu zabezpečené, že trénovacia sada bude obsahovať aspoň nejakú z pozorovaných hodnôt každého príznaku, čo by inak spôsobovalo problém imputačným metódam.

V prípade klasifikácie je ďalej pridané obmedzenie na povolený počet kategórií v cieľovom príznaku. Väčší počet kategórií by totiž by mohol spôsobiť nárast času potrebného na vytvorenie predikcií [35], čo je v aplikácii nežiadúce. Zabráni sa tým tiež problémom, ktoré vzniknú v dôsledku zvolenia spojitého príznaku pre klasifikáciu. V aktuálnej verzii je hraničná hodnota nastavená na 10, avšak v prípade záujmu je možné toto nastavenie zmeniť priradením inej hodnoty premennej `MAX_CLASS` v súbore `src/pages/page_data.py`.

Pre základné oboznámenie sa s dátovou sadou slúži stránka so štatistikami, ktorej ukážka je na obr. A.3 v prílohe. Zobrazuje tabuľku s jednotlivými príznakmi a ich relatívnym počtom chýbajúcich hodnôt v trénovacej a testovacej sade. Inak stránka pracuje len s trénovacou sadou. Výskyt chýbajúcich hodnôt v rámci nej je vizualizovaný pomocou maticového diagramu a dendrogramu. Na tejto stránke je ďalej graf korelačnej matice zobrazujúci Pearsonov korelačný koeficient medzi dvomi numerickými príznakmi. Okrem toho sú zobrazené popisné štatistiky jednotlivých príznakov. V prípade dátových typov `float` a `int` sú zobrazené údaje ako priemer, medián, štandardná odchýlka, minimum, maximum, či počet unikátnych hodnôt a tieto príznaky sú zobra-

zené histogramom. V prípade typu `category` je zobrazený počet unikátnych kategórií, najčastejšia kategória a tieto príznaky sú zobrazené stĺpcovým grafom.

6.2 Imputácia dát

Metódy z kapitoly 5 sú implementované na stránke s imputáciou dát. Sú sprevádzané teoretickými textami, ktoré vysvetľujú princíp na akom fungujú. Ukážka stránky s jednou z metód je na obr. A.4 v prílohe. Všetky metódy využívajú informácie dostupné len z trénovacej množiny pri dopĺňaní chýbajúcich hodnôt v trénovacej aj testovacej množine. Vo všetkých prípadoch je nastavený hyperparameter `random_state` kontrolujúci náhodnosť na hodnotu 42.

Metódy môžu byť rozšírené o iné vytvorením novej triedy. Je odporúčané, aby sa zakladala na šablóne `src/imputation/template_imputer.py`. Vytvorená trieda musí mať definované funkcie `provide_info()`, `param_tuning()`, `impute_data(data, int64_cols, target)`, `params_callback()` a definovaný atribút `name`. V súbore `src/pages/page_impute.py` je potom takto vytvorenú triedu potrebné importovať. V triede `ImputePage` je ďalej nutné aktualizovať atribúty `options` a `methods`. Podrobnosti toho, čo by ktorá funkcia mala robiť a čo jednotlivé atribúty znamenajú, možno nájsť priamo v príslušných súboroch implementácie.

Zmenu rozdelenia hodnôt príznakov v doplnenej dátovej sade v porovnaní s pôvodnou sadou možno pozorovať na stránke s výsledkami imputácií zobrazenej na obr. A.5 v prílohe. V prípade, že sa jedná o príznak dátového typu `int`, alebo `float` sú pre vizualizáciu použité histogramy. Ak ide o príznak typu `category`, sú v závislosti od počtu unikátnych kategórií použité buď stĺpcové grafy, alebo tabuľky. Vo všetkých prípadoch sú vizualizované len trénovacie množiny. Vykresľované grafy sú interaktívne a pri prejdení kurzorom ponad ne sa zobrazia konkrétne počty v danej kategórii, alebo v danom rozsahu hodnôt.

6.3 Trénovanie modelov

Predposledná stránka aplikácie je určená trénovaniu modelov strojového učenia. Na výber sú celkom tri v prípade klasifikácie aj regresie, a to rozhodovací strom, metóda podporných vektorov (SVM) a kNN. Všetky sú implementované využitím knižnice `scikit-learn`. Hyperparametre sú zvolené podľa predvolených nastavení s výnimkou rozhodovacieho stromu, kde je maximálna hĺbka stromu nastavená na 10. Inak častejšie dochádzalo k pretrénovaniu. Vo všetkých prípadoch sú nenumerické príznaky zakódované fiktívnymi príznakmi. Pri použití kNN sú navyše dáta normalizované pomocou min-max normalizácie. Rovnako ako pri imputácii je aj v tomto prípade nastavený hyperparameter `random_state` na hodnotu 42.

Modely sú súčasne natrénované na pôvodných a doplnených dátach. Pôvodné dáta sú nechané bez zmeny, ak dokáže zvolený algoritmus pracovať aj s chýbajúcimi hodnotami. Z implementovaných algoritmov tomu tak je len pri rozhodovacích stromoch. Inak sa chýbajúce hodnoty nahradia konštantou -1 a ku každému chýbajúcemu príznaku sa pridá nový binárny príznak indikujúci, či sa jednalo o chýbajúcu hodnotu.

Po natrénovaní modelov na trénovacích sadách sa porovnáva ich úspešnosť. Klasifikátory (viď obr. A.6 v prílohe) sa porovnávajú na základe percenta správnych predikcií. Ich výsledky sú vizuálne prezentované na testovacej sade pomocou normalizovanej matice zámen. Matica zámen obsahuje na x-ovej osi predikované kategórie a na y-ovej osi skutočné kategórie. Na jej primárnej diagonále sú potom správne predikované hodnoty.

Ak je navyše v prípade binárnej klasifikácie zjavné, ktorá z tried je pozitívna, zobrazená je aj krivka operačnej charakteristiky prijímača (ROC) a AUC. Krivka ROC zobrazuje súvislosť medzi vyšším počtom skutočne pozitívnych vzoriek za cenu vyššieho počtu falošne negatívnych vzoriek v závislosti na zmene hraničnej hodnoty, po ktorej prekročení je vzorka priradená pozitívnej triede. AUC nadobúda hodnoty od 0 do 1, kde 1 predstavuje ideálnu krivku ROC. [36]

Pri regresoroch (viď obr. A.7 v prílohe) sa modely porovnávajú na základe RMSE. Ich predikcie na testovacej množine sú porovnávané so skutočnými hodnotami histogramom.

Aplikácia môže byť rozšírená o iné modely vytvorením novej triedy. Je odporúčané, aby sa v prípade regresora zakladala na triede `RegressionTemplate` zo súboru `src/ml_training/regr_template.py` a v prípade klasifikátora na triede `ClassTemplate` zo súboru `src/ml_training/class_template.py`. Vytvorenej triede potom treba dodefinovať atribúty `model`, `model_imp`, `metric`, `metric_name`, `normalise` a `fill_orig`. Použitie modelu a vytvorenie výstupu je už implementované vo funkcii `train_model(train_page)`, ktorá by mala byť vhodná vo väčšine prípadov, ale je možné ju predefinovať aj inak. Vysvetlenie, čo jednotlivé atribúty znamenajú, možno nájsť v príslušných súboroch implementácie.

6.4 Zhrnutie výsledkov

Zhrnutie výsledkov všetkých vytvorených modelov strojového učenia podľa použitých imputačných metód je zobrazené na poslednej stránke, ktorej ukážka je na obr. A.8 v prílohe. Vykreslené sú celkom tri stĺpcové grafy. Prvé dva zobrazujú porovnanie modelov buď podľa percenta správnych predikcií, alebo podľa RMSE, v závislosti od typu riešeného problému. Sú v nich farebne odlíšené výsledky na trénovacej a testovacej množine. Prvý graf je pritom zoskupený podľa algoritmov strojového učenia použitých pre predikciu, teda ukazuje porovnanie rôznych metód doplnenia na rovnakých algoritmoch. Druhý je zoskupený podľa metód imputácie, teda ukazuje porovnanie rôznych algoritmov strojo-

vého učenia použitých pre predikciu pri použití rovnakých metód doplnenia. Oba grafy je možné usporiadať podľa názvu, alebo trénovacej, či testovacej presnosti. Posledný graf zobrazuje čas, ktorý imputačné metódy trvali. Všetky grafy sú interaktívne a po prejdení kurzorom ponad stĺpec grafu sa zobrazia použité hyperparametre metódy.

Kapitola 7
Experimenty

V rámci vytvorenej aplikácie bolo vykonaných niekoľko experimentov na reálnych dátových sadách. Niektoré z nich skúmali zmenu hodnôt po doplnení, iné porovnávali presnosť predikcií, sledovali vplyv zmeny hyperparametrov, alebo sa zamerali na dobu behu jednotlivých metód. Vo všetkých prípadoch sú výsledky prezentované na grafoch pochádzajúcich z aplikácie, ktorým boli upravené popisky.

7.1 Porovnanie hodnôt príznakov po doplnení

Prvý experiment je zameraný na porovnanie hodnôt príznakov po doplnení rôznymi metódami. Zvolená bola dátová sada [37] venujúca sa ochoreniu srdca. Používa sa pre predikciu prítomnosti choroby na základe určitých faktorov. Jedná sa teda o binárnu klasifikáciu.

Táto sada má celkom 1 025 záznamov a 14 príznakov. Neobsahuje žiadne chýbajúce hodnoty. Pre účely experimentu z nej bolo náhodne odstránených 30 % dátových bodov. Použité boli všetky implementované metódy imputácie s predvolenými hyperparametrami.

Zmeny hodnôt príznakov sú porovnávané graficky voči pôvodnej sade. Pozorované boli tri príznaky, a to vek, hladina cholesterolu v krvi a maximálna srdcová frekvencia. Všetky vizualizácie zobrazujú len trénovacie množiny. Modrou je zobrazovaný histogram príznaku v pôvodnej sade a oranžovou v doplnenej sade.

Porovnanie hodnôt veku je zobrazené na obr. 7.1. Je zjavné, že imputácia priemerom najmenej rešpektuje rozdelenie pôvodných hodnôt. Inak sú si histogramy príznaku po doplnení metódami MissForest, kNN, MICE aj GAIN podobné. V prípade GAIN sa ale ukazuje, že boli o niečo častejšie dopĺňané menej pozorované hodnoty v okolí priemeru, ktorý je približne 54.

Zmeny hodnôt cholesterolu vzniknuté doplnením sú zobrazené na obr. 7.2. Vo všetkých prípadoch boli dopĺňané najmä hodnoty 210–270. Pri použití

GAIN je histogram strmší. Dopĺňané hodnoty si boli viac podobné a pohybovali sa blízko priemeru, ktorý je približne 245.

Hodnoty maximálnej srdcovej frekvencie sú zobrazené na obr. 7.3. Výraznejšie zmeny, s výnimkou imputácie priemeru, však v tomto prípade nie sú pozorované.

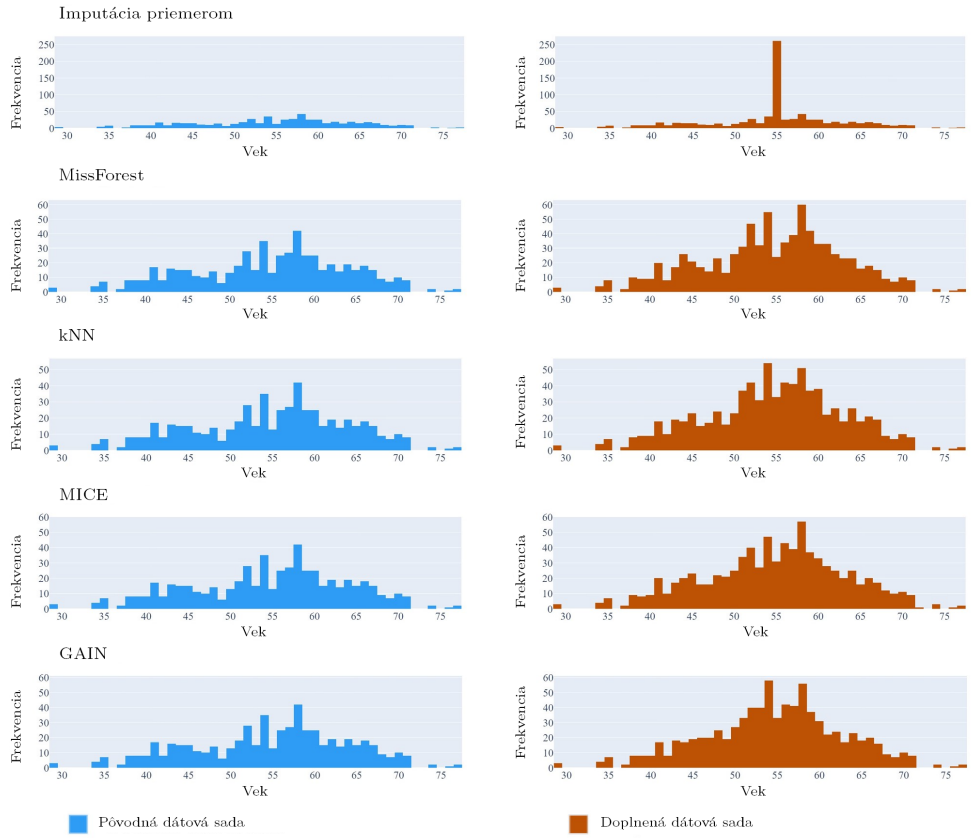
Celý experiment bol znovu zopakovaný pre 50 % náhodne odstránených bodov z celej dátovej sady. V takomto prípade je viac hodnôt, ktoré musia byť doplnené a tak sa môžu prejaviť väčšie rozdiely naprieč metódami.

Hodnoty získané po doplnení veku sú zobrazené na obr. 7.4. Metóda MissForest dopĺňala najmä hodnoty blízko 54 a 57. Metódy kNN, MICE a GAIN dopĺňali najviac hodnoty 45–60.

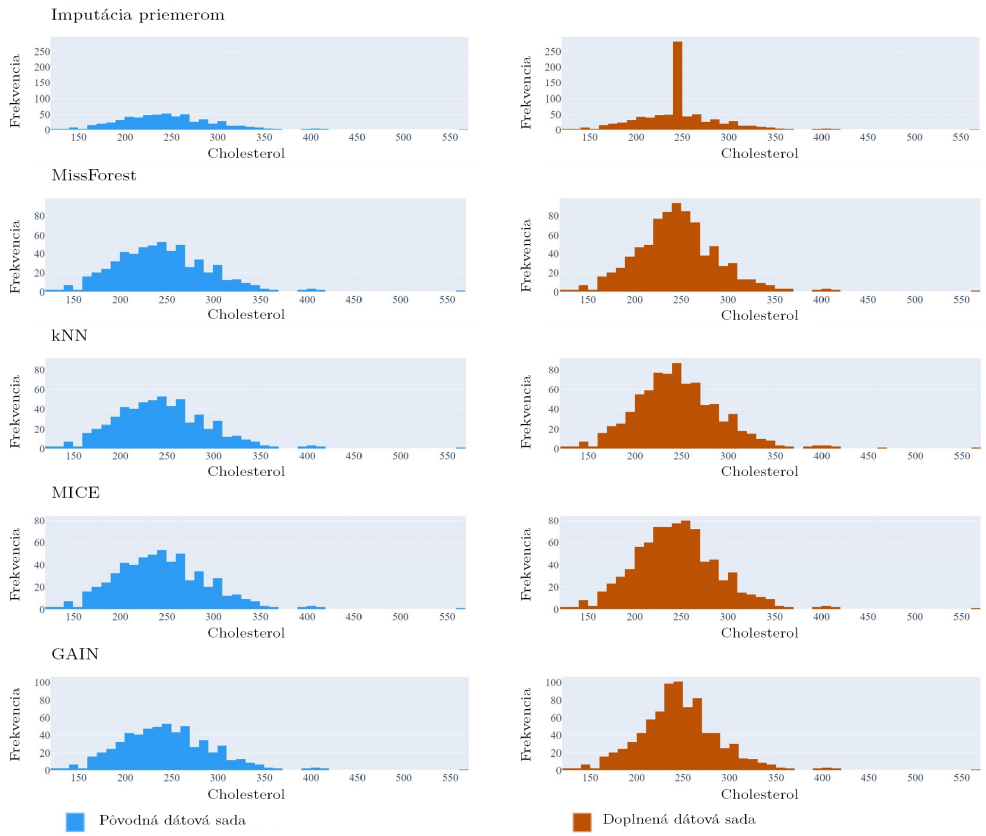
Porovnanie hodnôt cholesterolu je zobrazené na obr. 7.5. Hodnoty doplnené metódami kNN, MICE aj GAIN sú si podobné. Výraznejšie sa odlišujú hodnoty doplnené priemerom a metódou MissForest.

Pri pozorovaní maximálnej srdcovej frekvencie zobrazenej na obr. 7.6 nie sú ani v tomto prípade, s výnimkou imputácie priemeru, zistené významnejšie rozdiely.

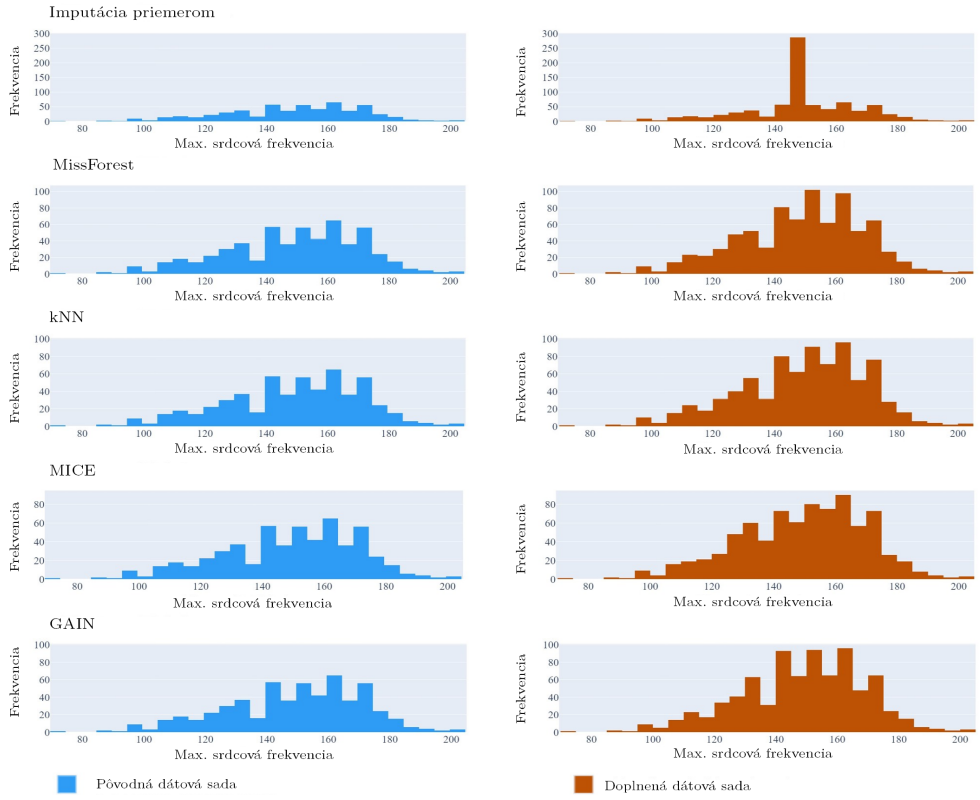
Ná základe experimentov sa teda ukazuje, že rôzne metódy nevplývajú na rozdelenie doplnených príznakov vždy rovnako, ale nimi dopĺňané hodnoty sa miestami líšia.



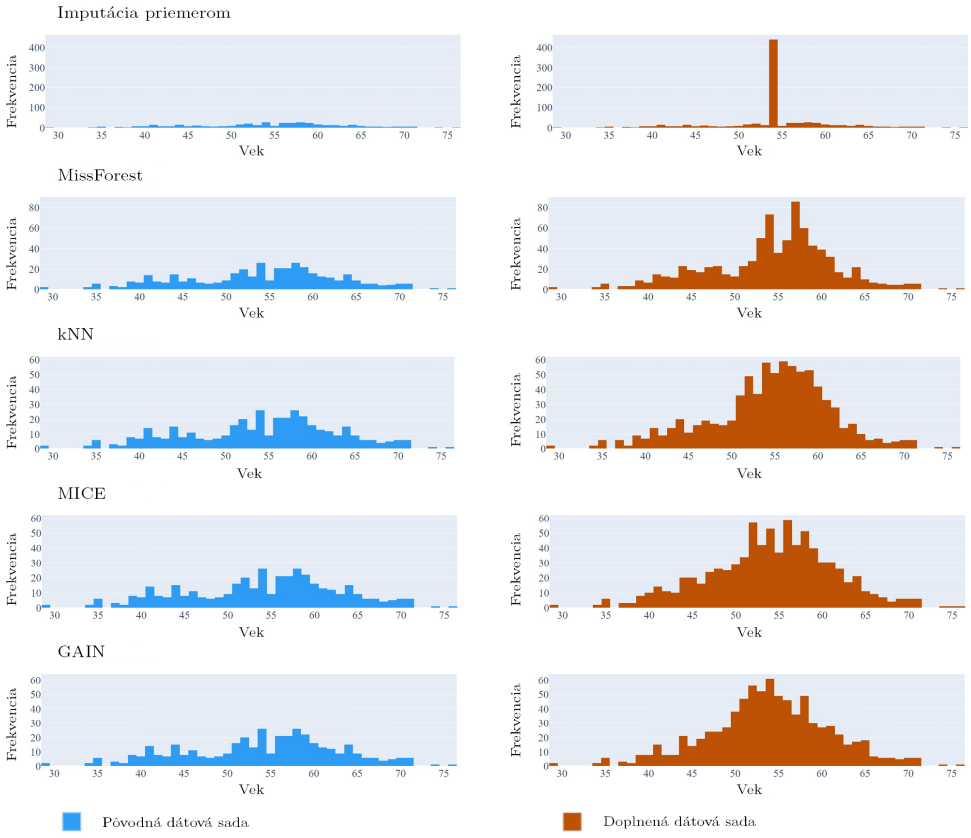
Obr. 7.1 Porovnanie zmien hodnôt veku po doplnení rôznymi metódami pri použití dátovej sady [37] s 30 % odstránených hodnôt



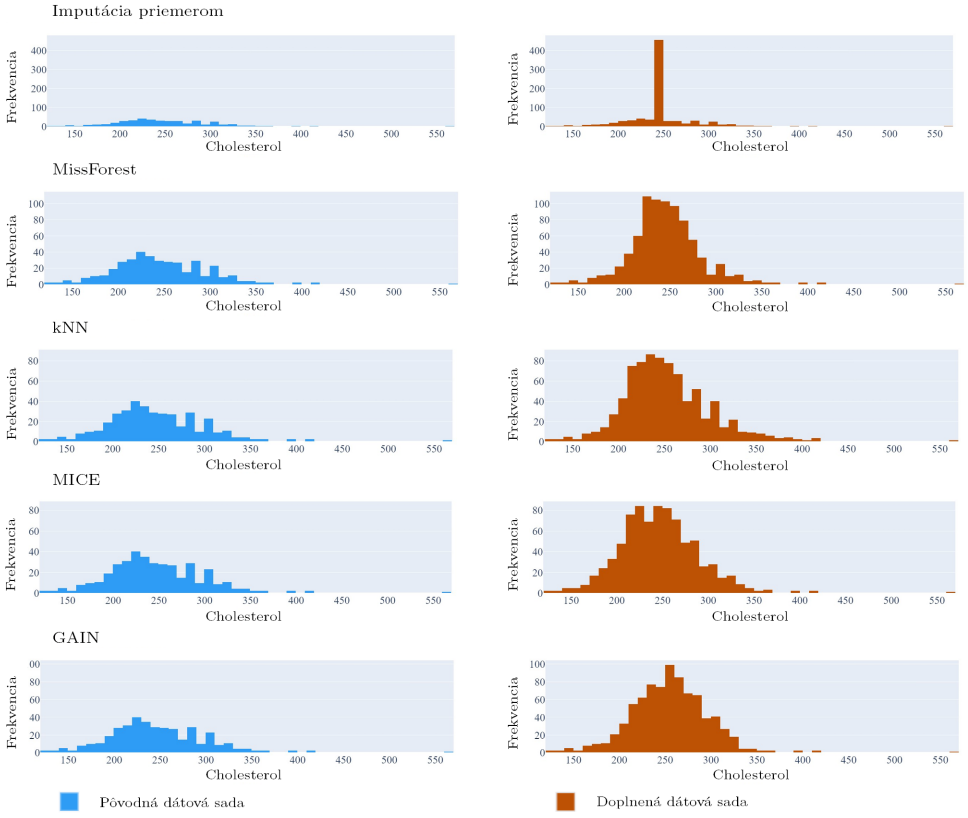
Obr. 7.2 Porovnanie zmien hodnôt cholesterolu po doplnení rôznymi metódami pri použití dátovej sady [37] s 30 % odstránených hodnôt



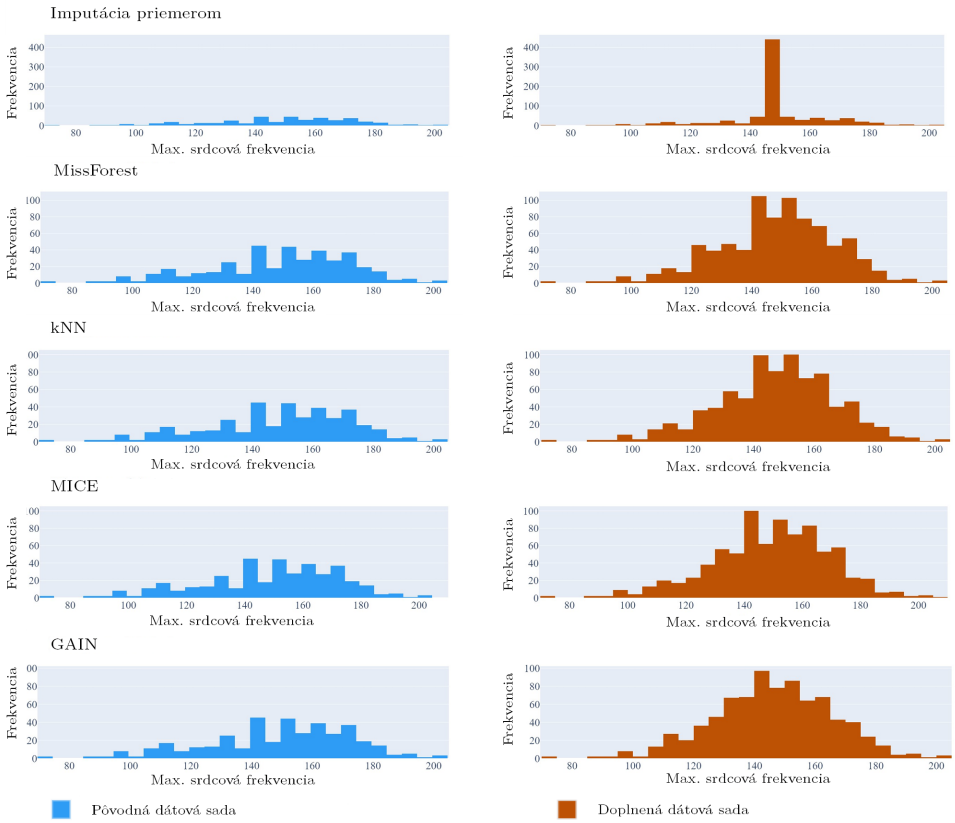
Obr. 7.3 Porovnanie zmien hodnôt srdcovej frekvencie po doplnení rôznymi metódami pri použití dátovej sady [37] s 30 % odstránených hodnôt



Obr. 7.4 Porovnanie zmien hodnôt veku po doplnení rôznymi metódami pri použití dátovej sady [37] s 50 % odstránených hodnôt



Obr. 7.5 Porovnanie zmien hodnôt cholesterolu po doplnení rôznymi metódami pri použití dátovej sady [37] s 50 % odstránených hodnôt



Obr. 7.6 Porovnanie zmien hodnôt srdcovej frekvencie po doplnení rôznymi metódami pri použití dátovej sady [37] s 50 % odstránených hodnôt

7.2 Vplyv hyperparametrov imputačných metód

Ďalší experiment sa zaoberá pozorovaním vplyvu hodnôt hyperparametrov imputačných metód na presnosť predikcií modelov strojového učenia.

Použitá dátová sada [38] má celkom 731 záznamov a 16 príznakov. Obsahuje informácie o zrážkach, vetre, teplote, či vlhkosti. Na základe týchto údajov je potom predikovaný počet požičaných bicyklov za jeden deň. Jedná sa teda o regresný problém.

Dátová sada neobsahuje žiadne chýbajúce hodnoty. Náhodne z nej bolo odstránených 30 % bodov. Príznak `dteday` bol pred experimentom odstránený, pretože informácie z neho sú v dátovej sade duplikované. Okrem toho bol ešte odstránený príznak `instant` označujúci index záznamu.

Ako prvá metóda bola zvolená MissForest. Boli v nej skúšané tri rôzne hodnoty maximálneho počtu iterácií, a to 5, 10 a 15. Na takto doplnené dáta boli potom aplikované všetky implementované algoritmy strojového učenia. Získané výsledky sú zobrazené na obr. 7.7, kde oranžová farba označuje testovaciu RMSE a modrá trénovaciu RMSE. Úspešnosť je porovnávaná na základe

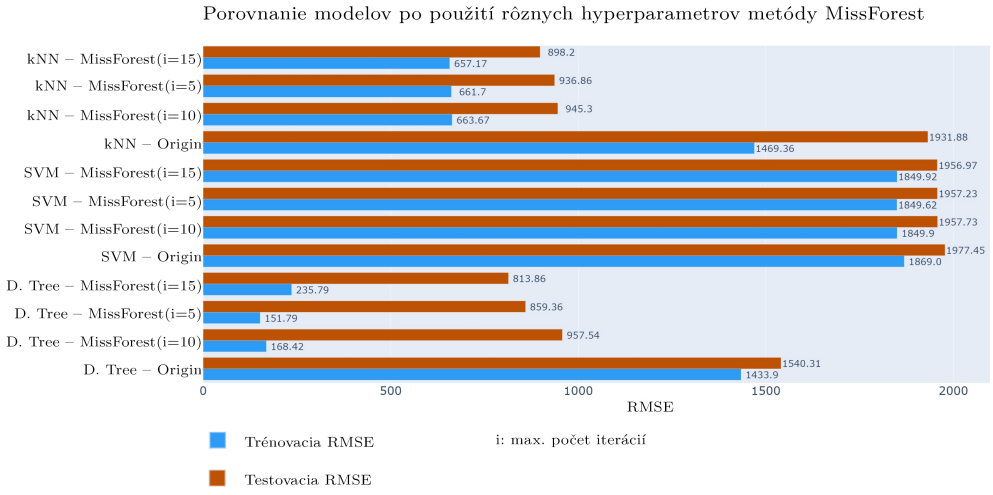
RMSE testovacej množiny. V prípade všetkých skúšaných algoritmov dopadlo najlepšie použitie hodnoty 15, potom 5 a nakoniec 10. Všetky skúšané hodnoty tohto hyperparametra zároveň viedli k lepším výsledkom než použitie pôvodného súboru s chýbajúcimi hodnotami.

Ďalšia použitá metóda bola MICE. Experimentovalo sa s počtom vytvorených kompletných dátových sád, pričom skúšané hodnoty boli 3, 5 a 7. Počet iterácií bol nastavený vo všetkých prípadoch na predvolenú hodnotu 10. Získané výsledky sú na obr. 7.8. Vo všetkých prípadoch sa ukázala byť najlepšou voľbou hodnota 7 a najhoršou voľbou hodnota 3. Všetky modely natrénované na dátach s doplnenými hodnotami dopadli lepšie než modely natrénované na pôvodných dátach.

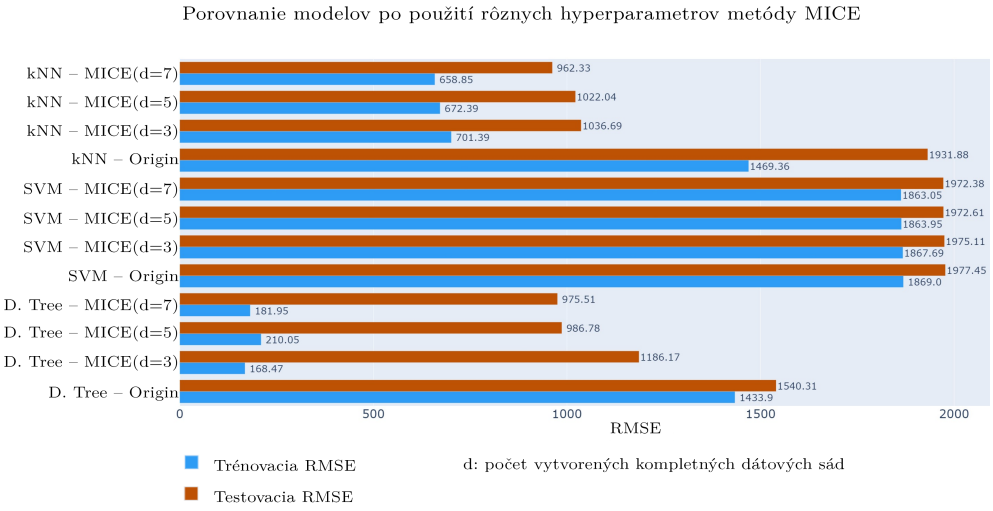
Pri použití metódy imputácie kNN možno ladiť počet susedov. Skúšané boli celkom tri hodnoty, a to 3, 5 a 7. Výsledky možno pozorovať na obr. 7.9. Modely kNN aj SVM natrénované na dátach doplnených metódou kNN dosiahli najnižšiu hodnotu RMSE pri 5 susedoch. Rozhodovací strom dosiahol najlepšie predikcie pri 7 susedoch. Aj v tomto prípade však použitie ľubovoľného počtu susedov dopadlo lepšie než použitie pôvodnej sady.

V metóde GAIN možno voliť veľkosť dávky, mieru nápovied, hyperparameter α a počet iterácií. Boli skúšané rôzne kombinácie hodnôt, ktoré sú zobrazené na obr. 7.10. Pri algoritme SVM dopadla najlepšie veľkosť dávky 256, hyperparameter α s hodnotou 100, počet iterácií 1 000 a miera nápovied 0,4. Pri kNN aj rozhodovacom strome to boli hodnoty 256 pri veľkosti dávky, 100 pri hyperparametri α, 1 200 pri počte iterácií a 0,8 pri miere nápovied. Použitie GAIN s akýmikoľvek skúšanými hyperparametrami viedlo k presnejším výsledkom než použitie pôvodnej sady.

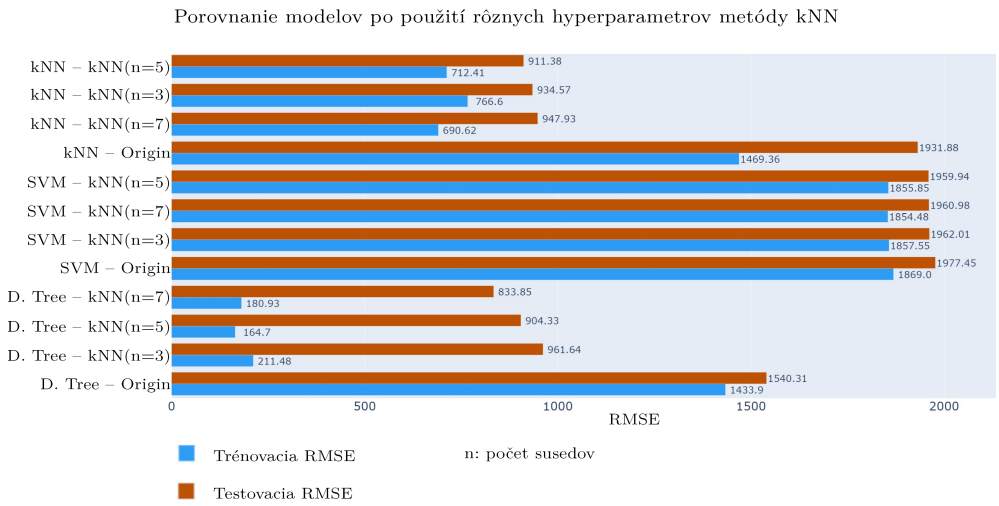
Na základe výsledkov sa ukazuje, že hyperparametre imputačných metód ovplyvňujú výsledné presnosti modelov natrénovaných na doplnených sadách.



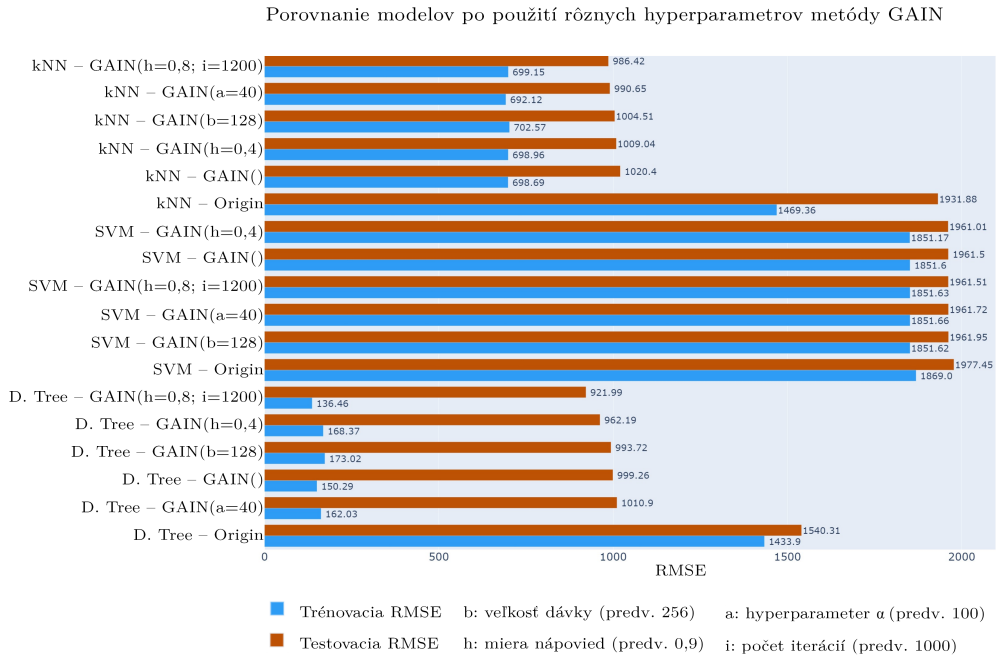
Obr. 7.7 Porovnanie modelov strojového učenia po doplnení dát metódou MissForest s rôznymi hyperparametrami za použitia dát [38] s 30 % odstránených hodnôt



Obr. 7.8 Porovnanie modelov strojového učenia po doplnení dát metódou MICE s rôznymi hyperparametrami za použitia dát [38] s 30 % odstránených hodnôt



Obr. 7.9 Porovnanie modelov strojového učenia po doplnení dát metódou kNN s rôznymi hyperparametrami za použitia dát [38] s 30 % odstránených hodnôt



Obr. 7.10 Porovnanie modelov strojového učenia po doplnení dát metódou GAIN s rôznymi hyperparametrami za použitia dát [38] s 30 % odstránených hodnôt

7.3 Vplyv objemu chýbajúcich hodnôt

Posledný experiment skúma vplyv objemu chýbajúcich hodnôt na presnosť predikcií modelov strojového učenia po použití rôznych metód imputácie. Pri tom tiež sleduje čas, ktorý jednotlivé doplnenia trvali.

Využitá dátová sada [39] obsahuje 569 záznamov a 33 príznakov. Jedná sa o reálnu dátovú sadu, ktorá sa používa pre predikciu typu nádoru prsníka na základe údajov získaných odberom. Cieľový príznak má dve kategórie: zhubný a nezhubný nádor.

Jeden z príznakov tejto sady má všetky hodnoty chýbajúce a pred vykonaním experimentov bol odstránený. Inak sú všetky hodnoty pozorované. Pre účely experimentu bolo z dátovej sady náhodne odstránených 10 %, 30 %, 50 % a 70 % hodnôt. Na konci bol ešte odstránený príznak `id` obsahujúci identifikačné číslo vzorky. Zvolené boli všetky imputačné metódy s predvolenými hyperparametrami a všetky algoritmy strojového učenia implementované v aplikácii.

Výsledky dosiahnuté po použití algoritmu kNN sú zobrazené na obr. 7.11. Oranžová farba označuje testovaciu presnosť a modrá trénovaciu presnosť. Grafy sú zoradené zostupne podľa testovacej presnosti a na základe nej sú tiež jednotlivé modely získané aplikovaním rôznych metód imputácie porovnávané. V prípade kNN dochádzalo s narastajúcim počtom chýbajúcich hodnôt k poklesu presnosti modelov trénovaných na pôvodnej sade. Pri použití metód imputácie boli pozorované len menšie rozdiely v prípade 10 %, 30 % a 50 % chýbajúcich bodov. Väčší pokles presnosti nastal až pri 70 % chýbajúcich bodov. Najvyššie dosiahnuté presnosti boli pri použití MICE a imputácie kNN, v závislosti od objemu chýbajúcich hodnôt. Vo všetkých prípadoch mal najnižšiu presnosť model trénovaný na pôvodných dátach. Aj imputácia priemerom teda výrazne zlepšila predikcie.

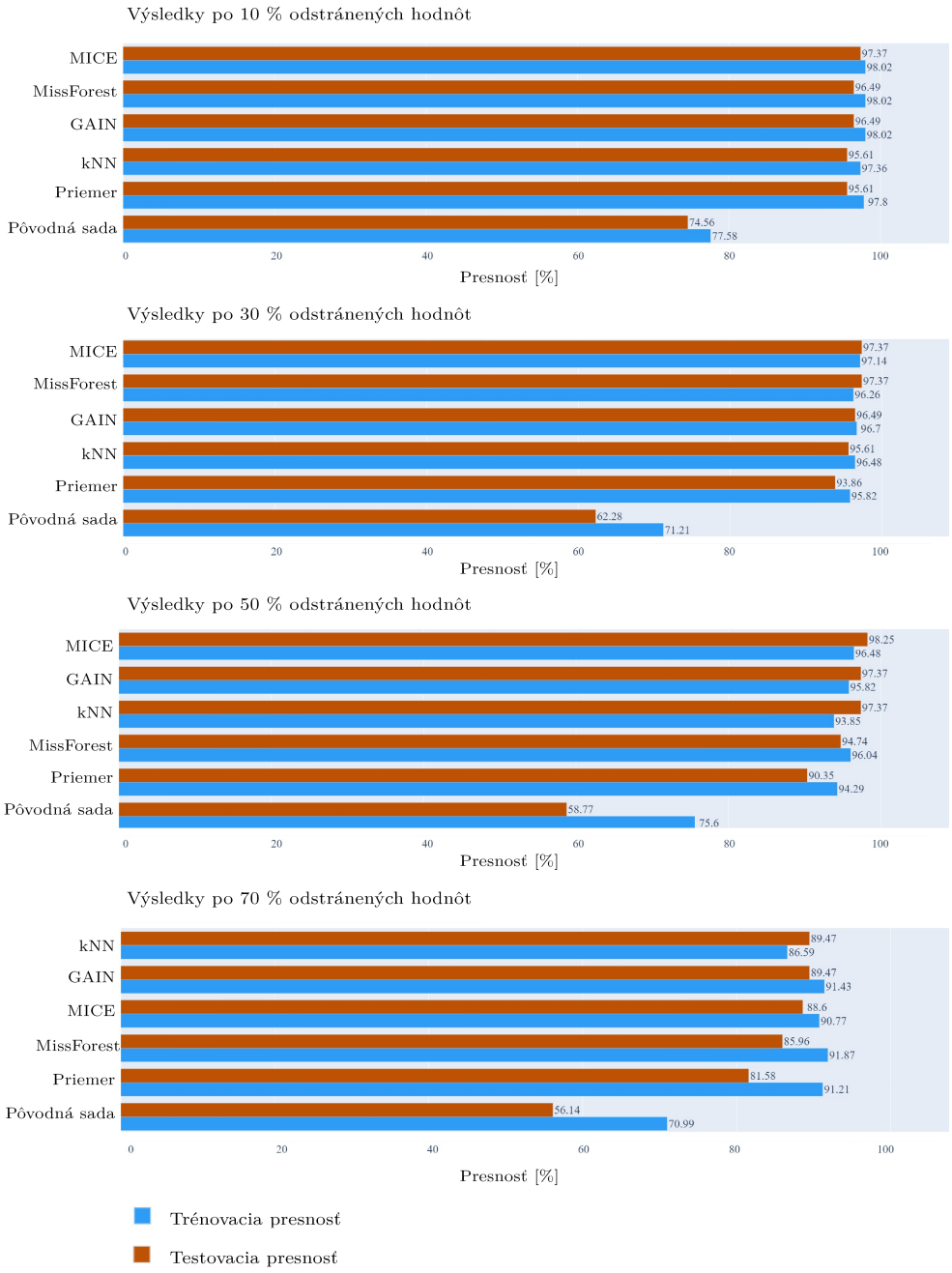
V prípade SVM, ktorej výsledky sú zobrazené na obr. 7.12, sa rovnako znižovala presnosť modelov trénovaných na pôvodnej sade s narastajúcim počtom chýbajúcich hodnôt. Rovnaký trend je pozorovaný aj pri imputácii priemerom. Pri použití ostatných metód doplnenia mali modely podobné výsledky v prípade 10 %, 30 % aj 50 % chýbajúcich bodov. Výraznejší pokles presnosti modelov bol opäť pozorovaný pri 70 %. Použitie metód GAIN, MICE a imputácie kNN viedlo k najlepším predikciám. V takmer všetkých prípadoch dopadla imputácia priemerom horšie ako ponechanie sady bez zmeny. Výnimkou je jej použitie pri 10 % chýbajúcich bodov. Ostatné metódy dopadli vždy lepšie, alebo porovnateľne dobre ako použitie pôvodnej sady.

Posledným použitým algoritmom je rozhodovací strom. Získané výsledky sú zobrazené na obr. 7.13. Vo väčšine prípadov sa s narastajúcim počtom chýbajúcich hodnôt znižovala jeho presnosť na pôvodnej aj doplnenej sade. Pri tomto algoritme sa výsledky použitia jednotlivých metód líšia výraznejšie v závislosti od objemu chýbajúcich hodnôt. Pri sade s 10 % chýbajúcich hodnôt dopadlo najlepšie doplnenie priemerom. Horšie ako použitie pôvodnej sady skončilo

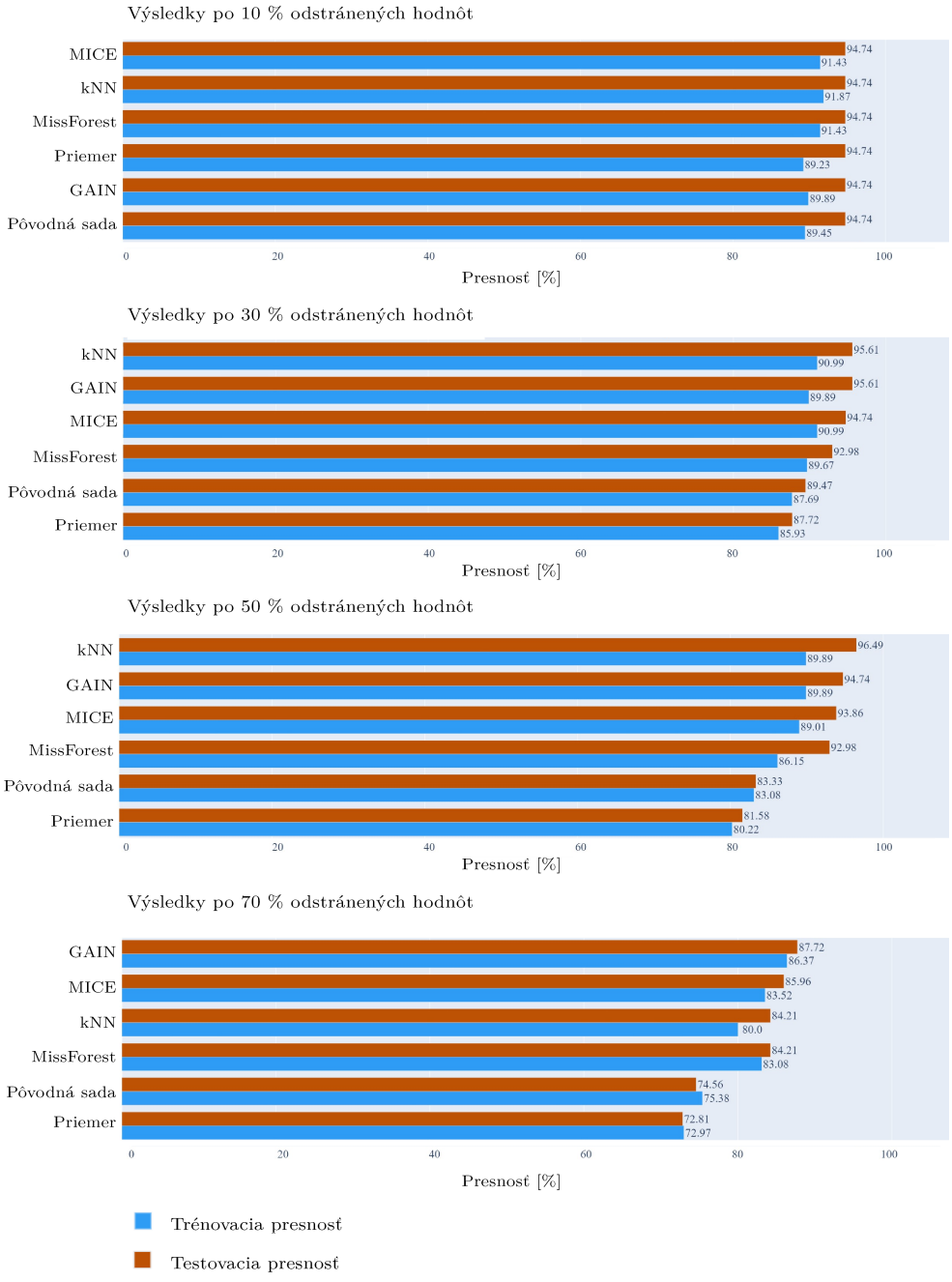
použitie sady doplnenej metódami MissForest a MICE. Pri 30 % chýbajúcich hodnôt dopadla najlepšie imputácia kNN. Naopak horšie než použitie pôvodnej sady dopadla imputácia priemerom. Pri 50 % chýbajúcich hodnôt boli najlepšie výsledky dosiahnuté po použití GAIN. Horšie ako použitie pôvodnej sady dopadlo doplnenie metódou MissForest. V prípade 70 % chýbajúcich hodnôt dosiahlo najlepšie výsledky použitie GAIN. Najhoršie dopadla pôvodná sada, imputácia priemerom a MissForest, pričom všetky tri mali rovnakú presnosť.

Na základe týchto výsledkov sa ukazuje, že voľba vhodnej metódy imputácie nezávisí len od dátovej sady a objemu chýbajúcich hodnôt, ale aj od zvoleného algoritmu strojového učenia používaného pre predikciu. Často však dopadlo doplnenie hodnôt ľubovoľnou metódou lepšie, než keď sa chýbajúce hodnoty nechali bez zmien.

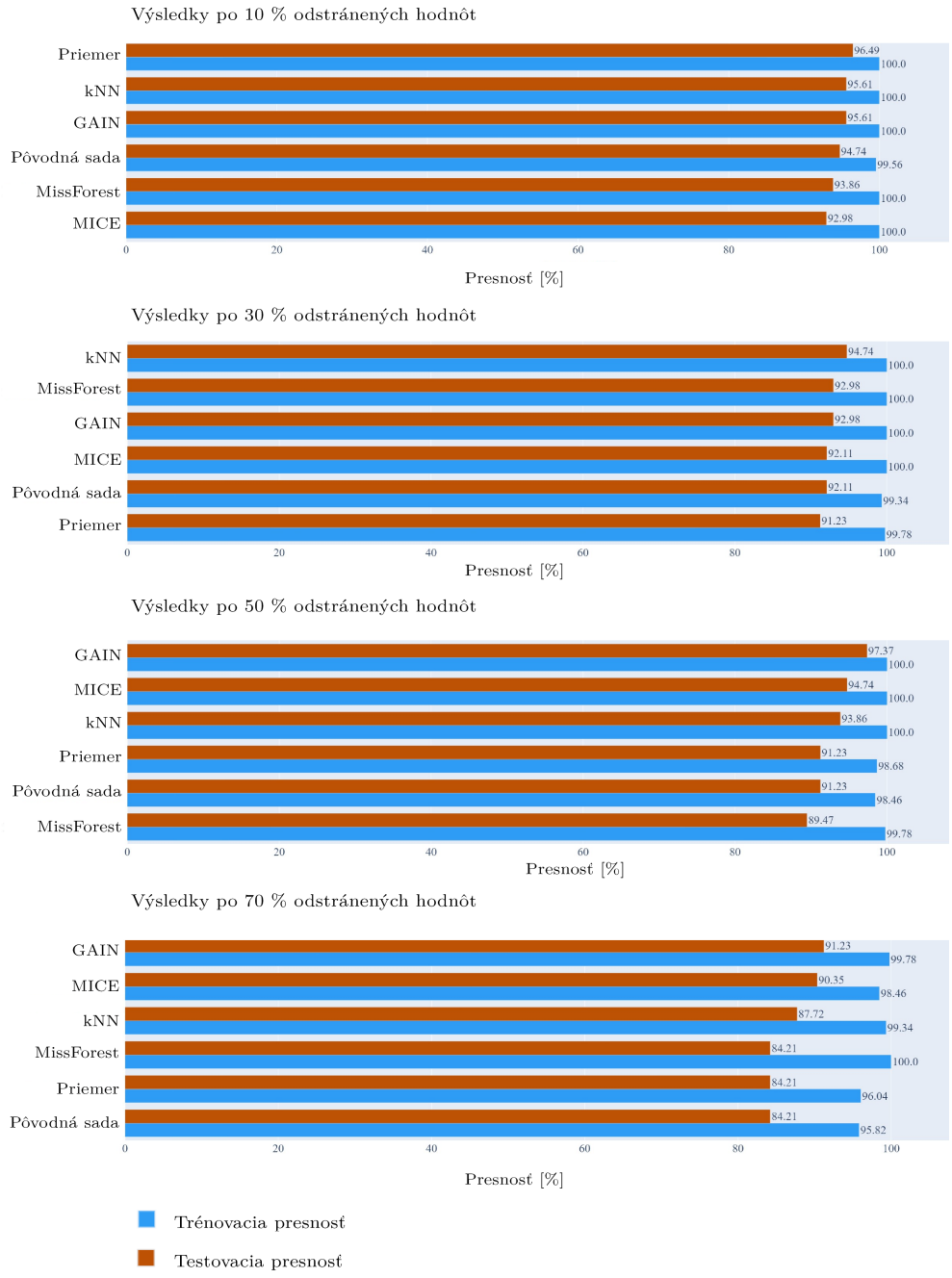
Doba behu použitých metód imputácie je postupne pre 10 %, 30 %, 50 % a 70 % chýbajúcich hodnôt zobrazená na obr. 7.14, 7.15, 7.16 a 7.17. Doba behu imputácie kNN mierne narastala so zvyšujúcim sa počtom chýbajúcich bodov. Čas potrebný pre imputáciu metódou MICE aj MissForest skôr klesal s narastajúcim počtom chýbajúcich bodov. Doba behu GAIN aj imputácie priemerom sa príliš nemenila so zmenou počtu bodov.



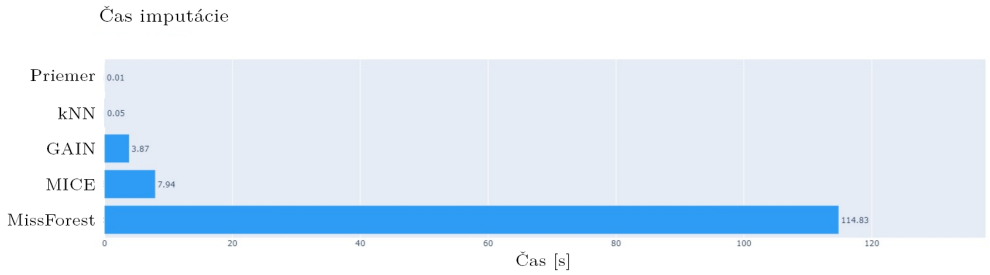
Obr. 7.11 Porovnanie presnosti algoritmu kNN po použití rôznych metód imputácie pri viacerých objemoch chýbajúcich hodnôt za použitia dátovej sady [39]



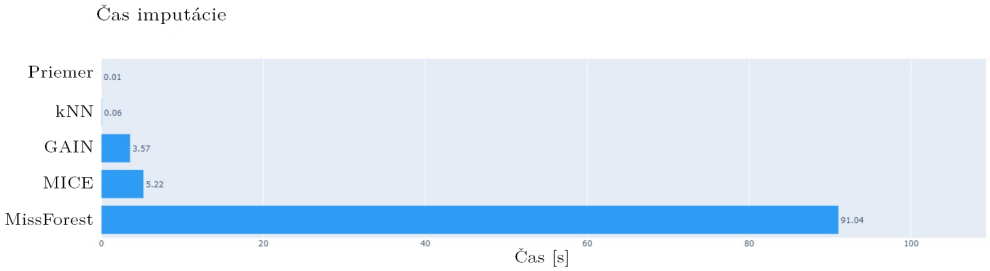
Obr. 7.12 Porovnanie presnosti algoritmu SVM po použití rôznych metód imputácie pri viacerých objemoch chýbajúcich hodnôt za použitia dátovej sady [39]



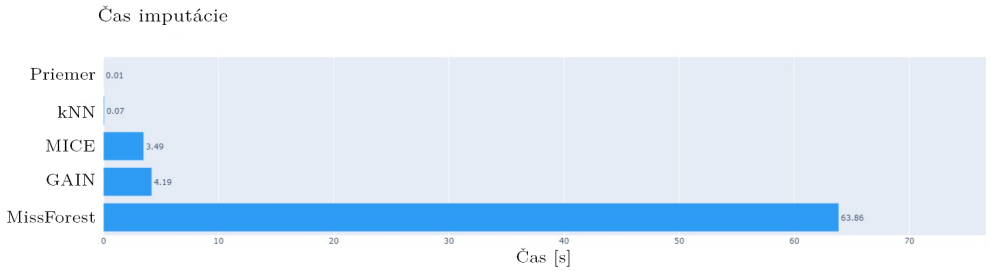
Obr. 7.13 Porovnanie presnosti rozhodovacieho stromu po použití rôznych metód imputácie pri viacerých objemoch chýbajúcich hodnôt za použitia dátovej sady [39]



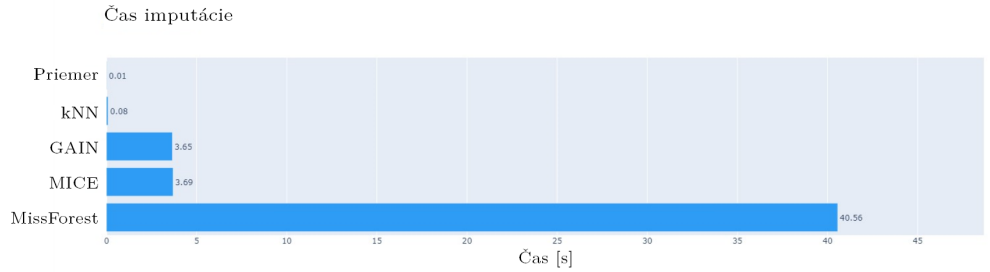
Obr. 7.14 Porovnanie trvania imputačných metód za použitia dát [39] s 10 % odstránených hodnôt



Obr. 7.15 Porovnanie trvania imputačných metód za použitia dát [39] s 30 % odstránených hodnôt



Obr. 7.16 Porovnanie trvania imputačných metód za použitia dát [39] s 50 % odstránených hodnôt



Obr. 7.17 Porovnanie trvania imputačných metód za použitia dát [39] so 70 % odstránených hodnôt

Kapitola 8
Záver

Hlavným cieľom tejto práce bolo navrhnúť a implementovať systém pre vizualizáciu chovania vybraných modelov strojového učenia vzhľadom k chýbajúcim dátam a ich doplneniu rôznymi metódami.

Výsledkom je webová aplikácia implementovaná v jazyku `Python`. Aplikácia ponúka možnosť nahrať dátovú sadu, prezrieť si jej riadky a stĺpce, či zobraziť štatistiky. Ďalej je možné sadu upraviť odstránením príznakov, alebo zmenou dátových typov. Následne je možné chýbajúce hodnoty doplniť implementovanými metódami. Na výber je celkom päť metód, a to imputácia priemerom, MissForest, imputácia kNN, MICE a GAIN. Voľbe týchto metód predchádzala rešerš, v rámci ktorej boli porovnávané výsledky rôznych štúdií zaoberajúcich sa danou problematikou. Po doplnení hodnôt je ďalej možnosť natrénovať na vzniknutých sadách vybrané modely strojového učenia. Implementované sú pre klasifikáciu aj regresiu tri, a to kNN, SVM a rozhodovací strom. Porovnáva sa vždy model natrénovaný na pôvodných a doplnených dátach. Na konci sú všetky výsledky použitých modelov a metód imputácie zhrnuté a reprezentované graficky. Aplikácia je zároveň vyvinutá tak, aby bolo možné jednoducho pridať aj ďalšie metódy imputácie a algoritmy strojového učenia používané pre predikciu.

Vytvorená aplikácia bola experimentálne otestovaná na troch reálnych dátových sadách, z ktorých boli náhodne odstránené rôzne objemy bodov. Ukázalo sa, že doplňané hodnoty príznakov sa naprieč rôznymi metódami líšia. V rámci experimentov mal často model natrénovaný na dátach po doplnení ľubovoľnou metódou presnejšie predikcie než model natrénovaný na pôvodných dátach s chýbajúcimi hodnotami. Rovnako bolo zistené, že aj voľba hyperparametrov imputačných metód dokáže ovplyvniť presnosť predikcií.

Vytvorená práca však nezahŕňa automatizovanejšie ladenie hyperparametrov. Zaujímavým rozšírením by mohlo byť implementovanie tzv. vyhľadávania v mriežke (z angl. *grid search*), ktoré skúša ich rôzne kombinácie.

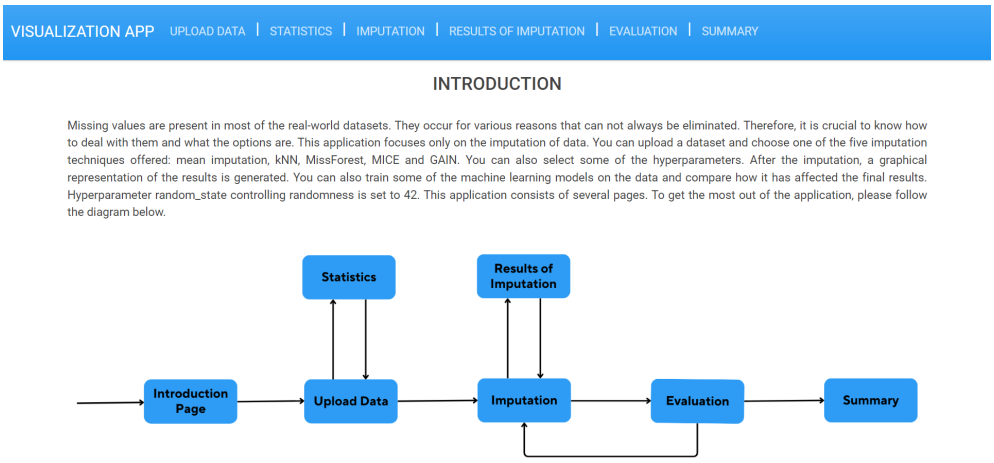
V rámci tejto aplikácie je tiež prednastavený hyperparameter kontrolujúci náhodnosť na číslo 42. V budúcnosti by mohla byť aplikácia rozšírená o mož-

nosť toto nastavenie odstrániť a získané výsledky exportovať formou tabuľky. Takáto funkcionalita by bola vhodná najmä pre účely ďalšieho štatistického spracovania.

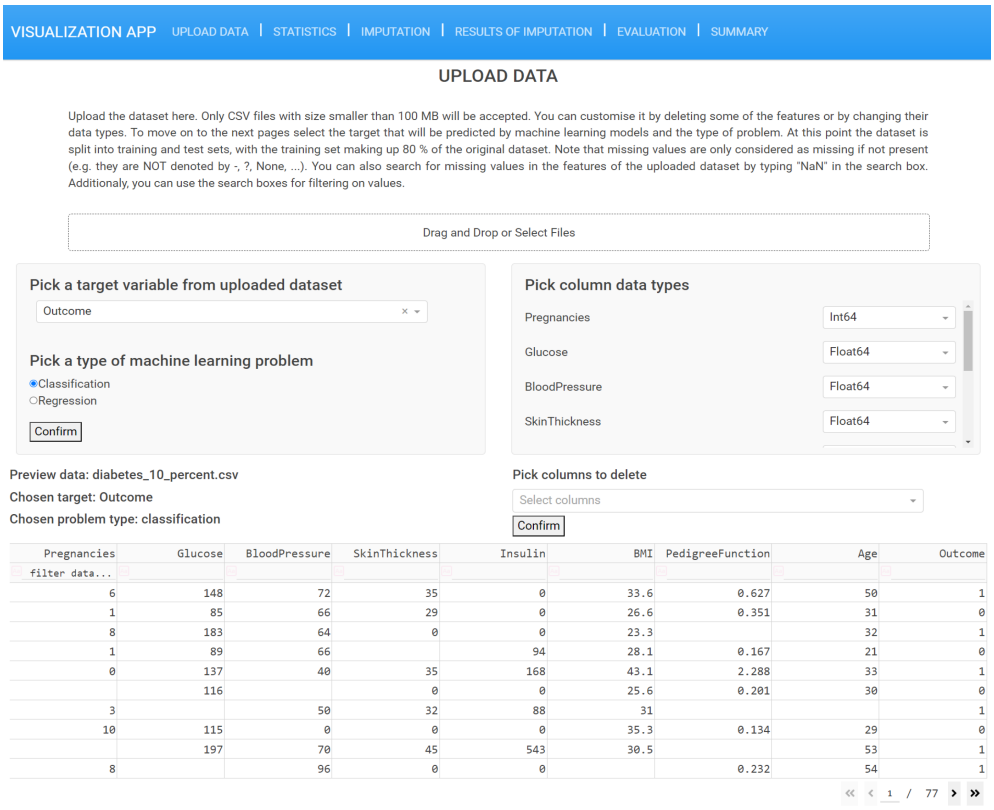
Aplikácia v súčasnej verzii slúži najmä pre oboznámenie sa s danými metódami imputácie a pozorovanie vplyvu ich použitia na výsledky predikcií modelov strojového učenia prostredníctvom vizualizácií.

Dodatok A

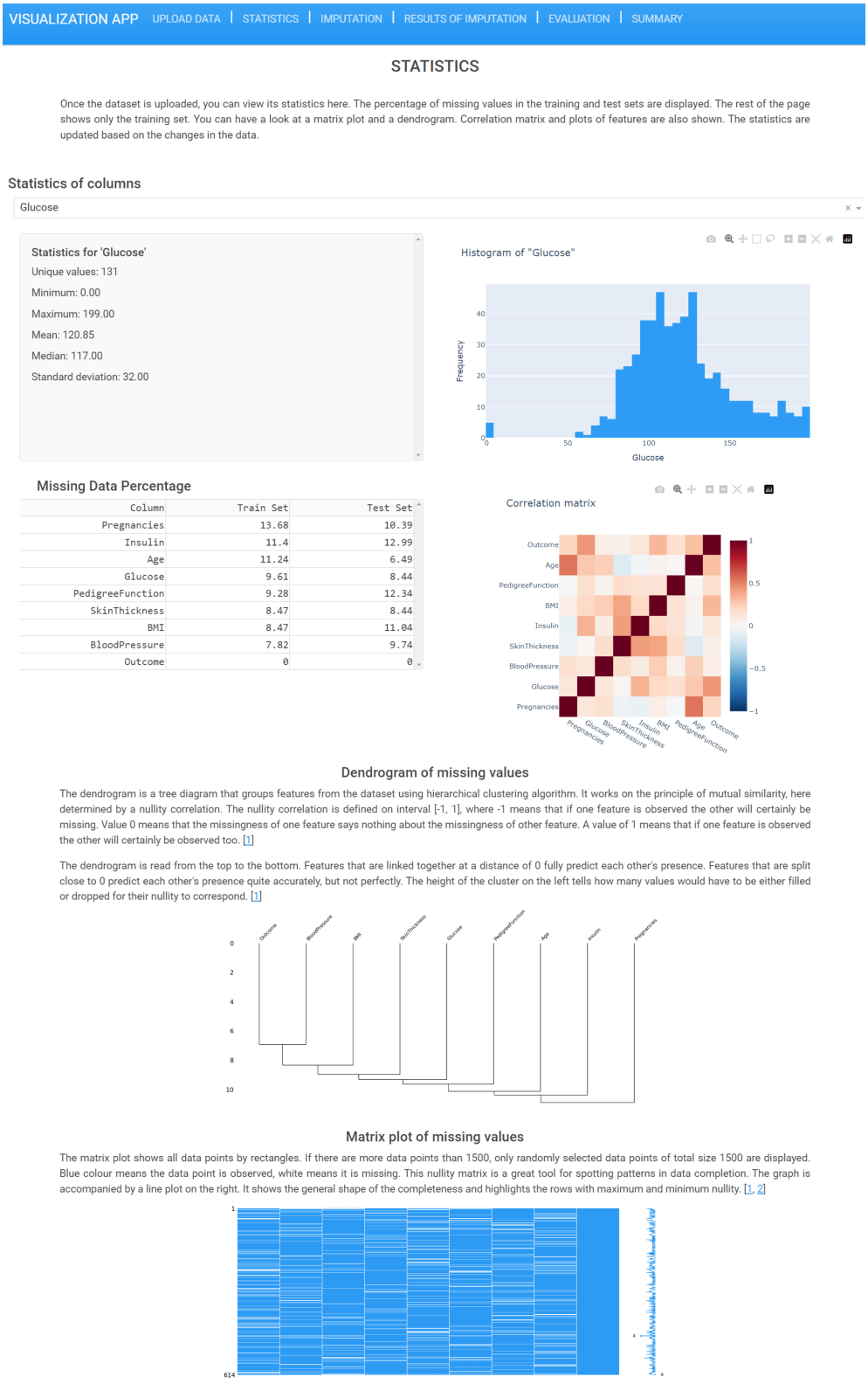
Ukážka stránok aplikácie



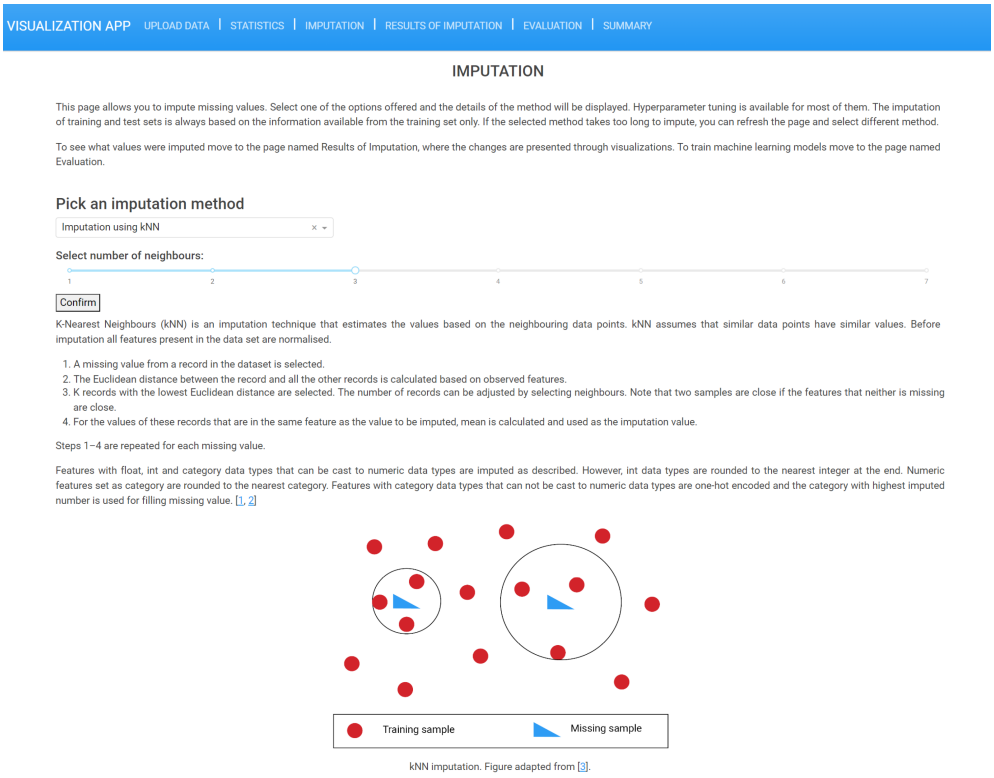
Obr. A.1 Ukážka úvodnej stránky aplikácie



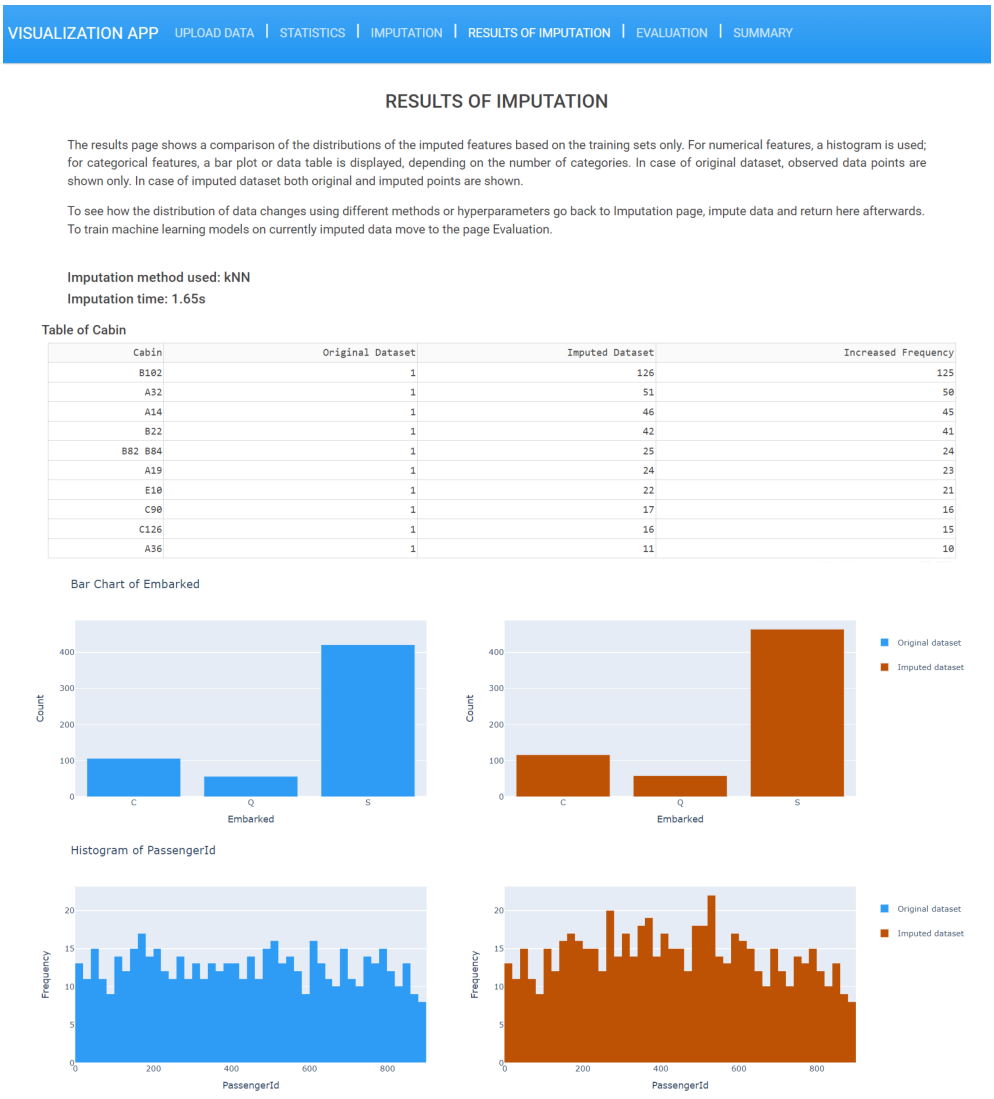
Obr. A.2 Ukážka stránky aplikácie s nahrávaním dát



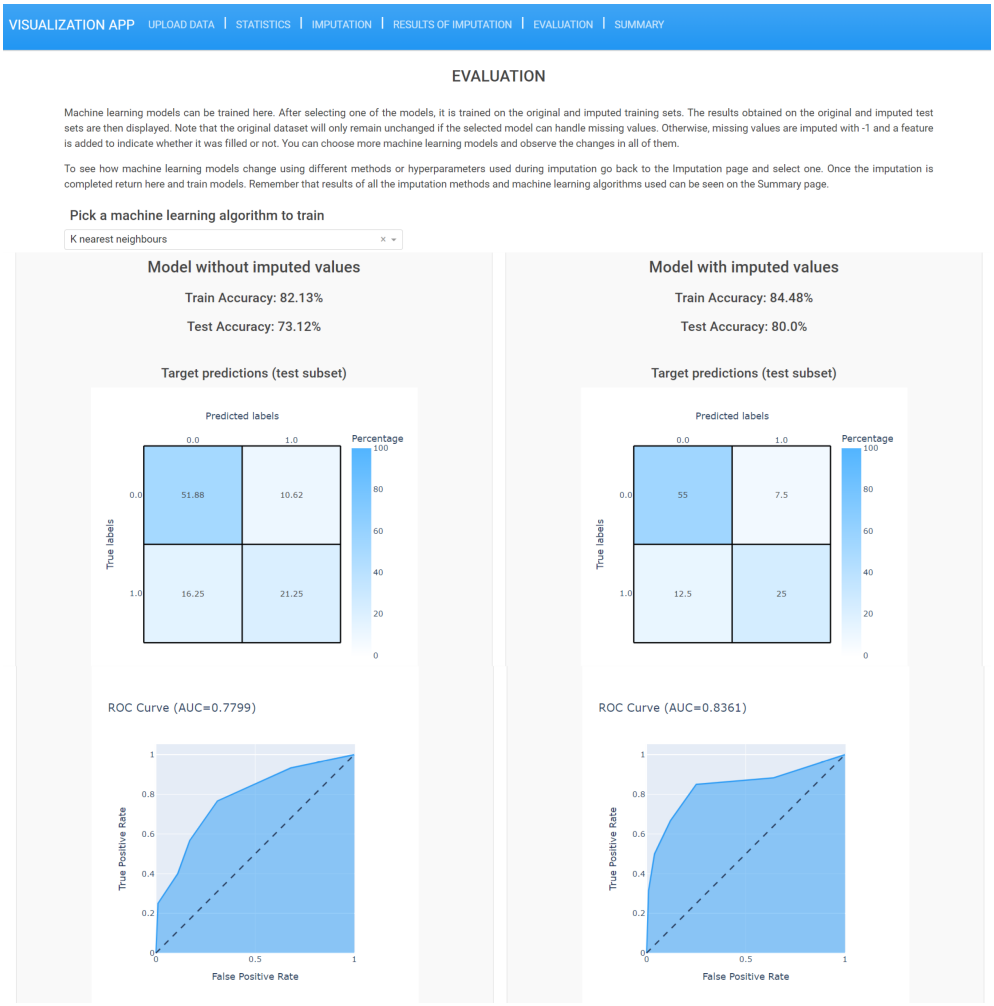
Obr. A.3 Ukážka stránky aplikácie so štatistikami



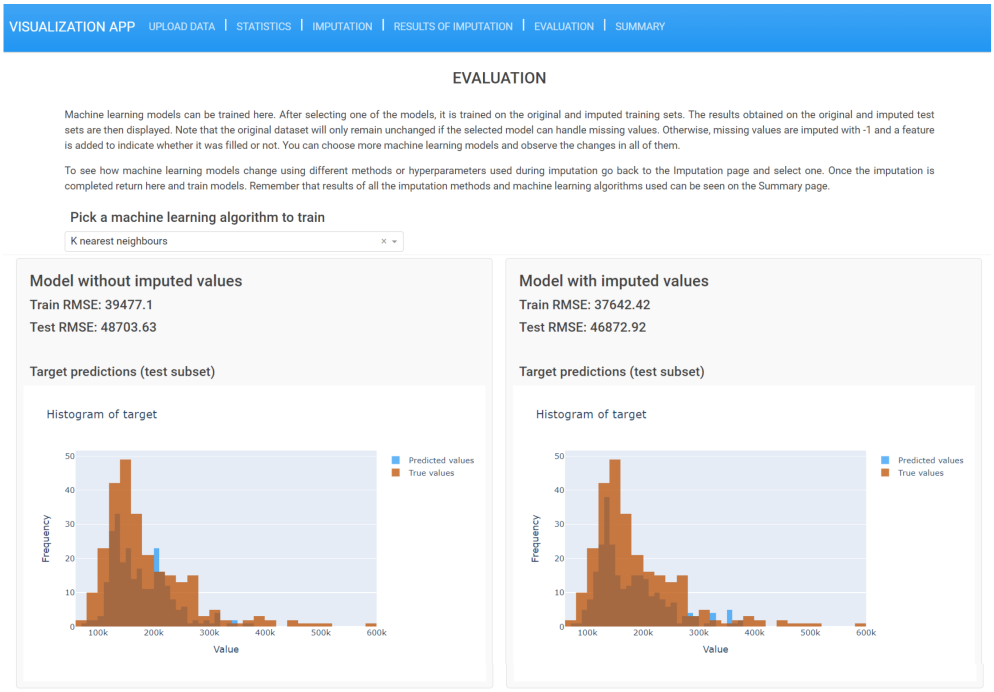
Obr. A.4 Ukážka stránky aplikácie s imputáciou



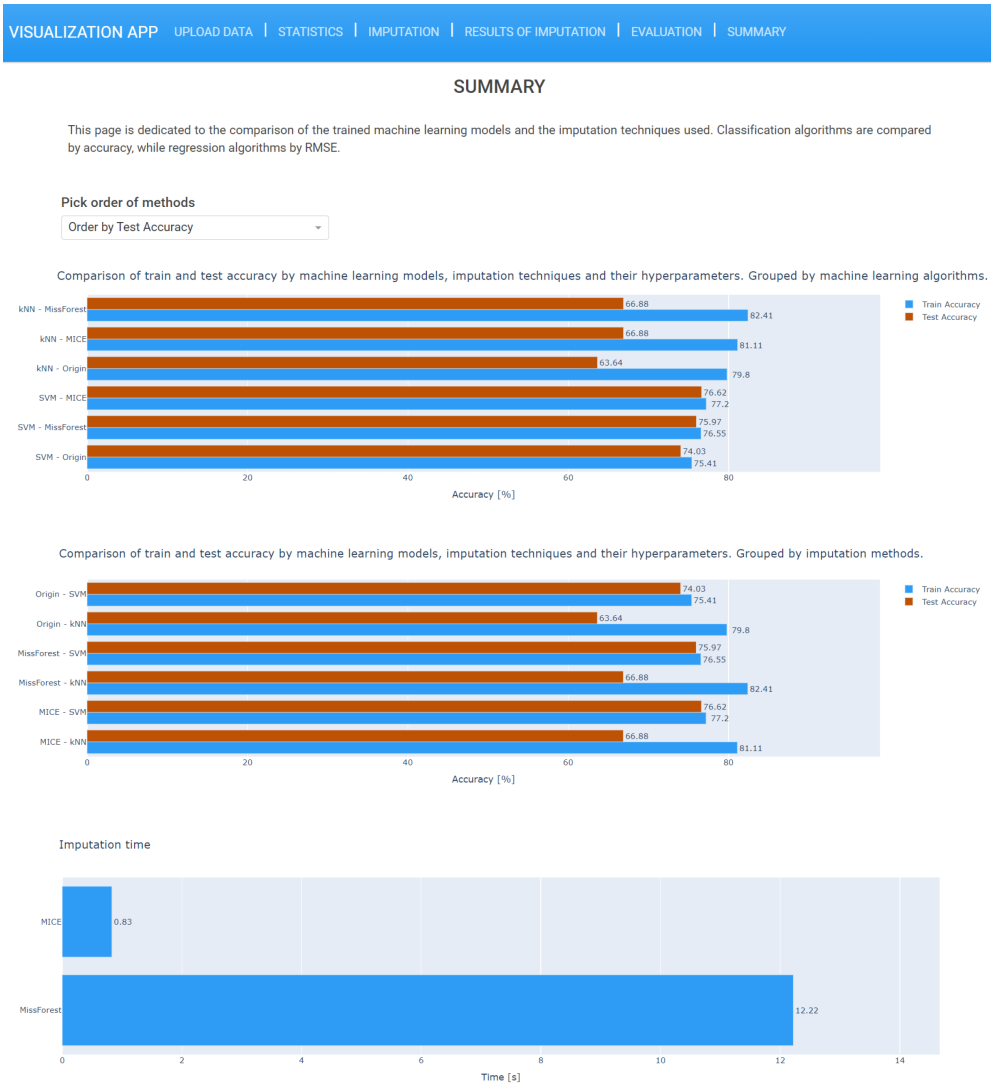
Obr. A.5 Ukážka stránky aplikácie s výsledkami imputácie



Obr. A.6 Ukážka stránky aplikácie s porovnaním natrénovaných klasifikátorov



Obr. A.7 Ukážka stránky aplikácie s porovnaním natrénovaných regresorov



Obr. A.8 Ukážka stránky aplikácie so zhrnutím presnosti natrénovaných modelov za použitia rôznych imputačných metód

Bibliografia

1. JÄGER, Sebastian; ALLHORN, Arndt; BIESSMANN, Felix. A Benchmark for Data Imputation Methods. *Frontiers in Big Data*. 2021, roč. 4. ISSN 2624-909X. Dostupné z DOI: 10.3389/fdata.2021.693674.
2. BILOGUR, Aleksey. Missingno: a missing data visualization suite. *The Journal of Open Source Software*. 2018, roč. 3, s. 547. Dostupné z DOI: 10.21105/joss.00547.
3. JOCH, Otakar. *Vizualizace chování algoritmů strojového učení vzhledem k charakteru datových sad*. Brno, Česká republika, 2020. Bakalárska práca. Masarykova univerzita. Dostupné z : https://is.muni.cz/th/yhc2f/Plny_text_prace.pdf.
4. MURPHY, Kevin P. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020. Dostupné tiež z: https://books.google.sk/books?id=NZP6AQAAQBAJ.
5. KANG, Hyun. The prevention and handling of the missing data. *Korean Journal of Anesthesiology*. 2013, roč. 64, č. 5, s. 402–406. Dostupné z DOI: 10.4097/kjae.2013.64.5.402.
6. BUUREN, Stef van. *Flexible Imputation of Missing Data, Second Edition*. Second Edition. Chapman a Hall/CRC, 2018. Dostupné z DOI: 10.1201/9780429492259.
7. NAHHAS, Ramzi W. Introduction to Regression Methods for Public Health Using R. *Bookdown* [online]. 2023 [cit. 2023-09-02]. Dostupné z : https://bookdown.org/rwnahhas/RMPH/.
8. SCHAFER, Joseph L.; GRAHAM, John W. Missing data: our view of the state of the art. *Psychological methods*. 2002, roč. 7, s. 147–177. Dostupné z DOI: 10.1037/1082-989X.7.2.147.

9. TEMPL, Matthias; ALFONS, Andreas; FILZMOSER, Peter. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*. 2012, roč. 6, s. 29–47. Dostupné z DOI: 10.1007/s11634-011-0102-y.

10. BILOGUR, Aleksey. Missing data visualization module for Python. *GitHub* [online]. 2018 [cit. 2024-04-04]. Dostupné z : https://github.com/ResidentMario/missingno.

11. STAVSETH, Marianne Riksheim; CLAUSEN, Thomas; RØISLIEN, Jo. How handling missing data may impact conclusions: A comparison of six different imputation methods for categorical questionnaire data. *SAGE Open Medicine*. 2019, roč. 7. Dostupné z DOI: 10/gf85jb.

12. LITTLE, Roderick; RUBIN, Donald. *Statistical Analysis with Missing Data, Third Edition*. Wiley, 2019. Wiley Series in Probability and Statistics. ISBN 9780470526798. Dostupné z DOI: 10.1002/9781119482260.

13. SALGADO, Cátia M.; AZEVEDO, Carlos; PROENÇA, Hugo; VIEIRA, Susana M. Missing Data. In: *Secondary Analysis of Electronic Health Records*. Cham: Springer International Publishing, 2016, s. 143–162. ISBN 978-3-319-43742-2. Dostupné z DOI: 10.1007/978-3-319-43742-2_13.

14. LAROSE, Daniel T.; LAROSE, Chantal D. Imputation of Missing Data. In: *Discovering Knowledge in Data*. John Wiley & Sons, Ltd, 2014, kap. 13, s. 266–276. ISBN 9781118874059. Dostupné z DOI: https://doi.org/10.1002/9781118874059.ch13.

15. JEREZ, José M. a kol. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*. 2010, roč. 50, č. 2, s. 105–115. ISSN 0933-3657. Dostupné z DOI: https://doi.org/10.1016/j.artmed.2010.05.002.

16. BUUREN, Stef van; GROOTHUIS-OUDSHOORN, Karin. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*. 2011, roč. 45, č. 3, s. 1–67. Dostupné z DOI: 10.18637/jss.v045.i03.

17. HASAN, Md. Kamrul a kol. Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Informatics in Medicine Unlocked*. 2021, roč. 27, s. 100–799. ISSN 2352-9148. Dostupné z DOI: https://doi.org/10.1016/j.imu.2021.100799.

18. SUN, Yige; LI, Jing; XU, Yifan; ZHANG, Tingting; WANG, Xiaofeng. Deep learning versus conventional methods for missing data imputation: A review and comparative study. *Expert Systems with Applications*. 2023, roč. 227. ISSN 0957-4174. Dostupné z DOI: https://doi.org/10.1016/j.eswa.2023.120201.

19. DONG, Weinan a kol. Generative adversarial networks for imputing missing data for big data clinical research. *BMC medical research methodology*. 2021, roč. 21, s. 1–10. Dostupné z DOI: 10.1186/s12874-021-01272-3.

20. PLATIAS, Christos; PETASIS, Georgios. A Comparison of Machine Learning Methods for Data Imputation. In: *11th Hellenic Conference on Artificial Intelligence*. Athens, Greece: Association for Computing Machinery, 2020, s. 150–159. SETN 2020. ISBN 9781450388788. Dostupné z DOI: 10.1145/3411408.3411465.

21. STEKHOVEN, Daniel J.; BÜHLMANN, Peter. MissForest–non-parametric missing value imputation for mixed-type data. *Bioinformatics*. 2011, roč. 28, č. 1, s. 112–118. ISSN 1367-4803. Dostupné z DOI: 10.1093/bioinformatics/btr597.

22. HUNT, Lynette A. Missing Data Imputation and Its Effect on the Accuracy of Classification. In: PALUMBO, Francesco; MONTANARI, Angela; VICHI, Maurizio (ed.). *Data Science*. Cham: Springer International Publishing, 2017, s. 3–14. ISBN 978-3-319-55723-6. Dostupné z DOI: https://doi.org/10.1007/978-3-319-55723-6_1.

23. YOON, Jinsung; JORDON, James; SCHAAR, Mihaela van der. GAIN: Missing Data Imputation using Generative Adversarial Nets. In: DY, Jennifer; KRAUSE, Andreas (ed.). *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018, zv. 80, s. 5689–5698. Proceedings of Machine Learning Research. Dostupné tiež z: https://proceedings.mlr.press/v80/yoon18a.html.

24. FRIEDJUNGOVÁ, Magda; VAŠATA, Daniel; BALATSKO, Maksym; JIŘINA, Marcel. Missing Features Reconstruction Using a Wasserstein Generative Adversarial Imputation Network. *CoRR*. 2020, roč. abs/2006.11783. Dostupné z DOI: 10.1007/978-3-030-50423-6_17.

25. KHENNOU, Fadoua; FAHIM, Charif; CHAOUI, Habiba; NOUR EL HOUDA, Chaoui. A Machine Learning Approach: Using Predictive Analytics to Identify and Analyze High Risks Patients with Heart Disease. *International Journal of Machine Learning and Computing*. 2019, roč. 9, s. 762–767. Dostupné z DOI: 10.18178/ijmlc.2019.9.6.870.

26. PEDREGOSA, F. a kol. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* [online]. 2011, roč. 12, s. 2825–2830 [cit. 2024-02-06]. Dostupné z : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.nan_euclidean_distances.html.

27. PEUGH, James L.; ENDERS, Craig K. Missing Data in Educational Research: A Review of Reporting Practices and Suggestions for Improvement. *Review of Educational Research*. 2004, roč. 74, č. 4, s. 525–556. Dostupné z DOI: 10.3102/00346543074004525.

28. AZUR, Melissa J.; STUART, Elizabeth A.; FRANGAKIS, Constantine; LEAF, Philip J. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*. 2011, roč. 20, č. 1, s. 40–49. Dostupné z DOI: 10.1002/mpr.329.

29. HEYMANS, Martijn W; EEKHOUT, Iris. Applied missing data analysis with SPSS and (R) Studio. *Bookdown* [online]. 2019 [cit. 2024-04-25]. Dostupné z : https://bookdown.org/mwheymans/bookmi/.

30. MERA-GAONA, Maritza; NEUMANN, Ursula; VARGAS, Rubiel; LÓPEZ, Diego. Evaluating the impact of multivariate imputation by MICE in feature selection. *PLOS ONE*. 2021, roč. 16. Dostupné z DOI: 10.1371/journal.pone.0254720.

31. UDILĂ, Andrei. *Encoding methods for categorical data: A comparative analysis for linear models, decision trees, and support vector machines*. Delft, Holandsko, 2023. Bakalárska práca. Delft University of Technology. Dostupné z : http://resolver.tudelft.nl/uuid:10b91b99-2685-4a45-b44e-48fbbf808ce2.

32. SHAHBAZIAN, Reza; TRUBITSYNA, Irina. DEGAIN: Generative-Adversarial-Network-Based Missing Data Imputation. *Information*. 2022, roč. 13, č. 12. ISSN 2078-2489. Dostupné z DOI: 10.3390/info13120575.

33. YAN, Ming; CHEN, Junjie; ZHANG, Xiangyu; TAN, Lin; WANG, Gan; WANG, Zan. Exposing numerical bugs in deep learning via gradient backpropagation. In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Athens, Greece: Association for Computing Machinery, 2021, s. 627–638. ESEC/FSE 2021. ISBN 9781450385626. Dostupné z DOI: 10.1145/3468264.3468612.

34. MA, Jianghong; CHOW, Tommy W.S. Label-specific feature selection and two-level label recovery for multi-label classification with missing labels. *Neural Networks*. 2019, roč. 118, s. 110–126. ISSN 0893-6080. Dostupné z DOI: https://doi.org/10.1016/j.neunet.2019.04.011.

35. BENGIO, Samy; WESTON, Jason; GRANGIER, David. Label embedding trees for large multi-class tasks. In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems – Volume 1*. Vancouver, British Columbia, Canada: Curran Associates Inc., 2010, s. 163–171. NIPS'10. Dostupné tiež z: https://api.semanticscholar.org/CorpusID:312693.

36. FOGARTY, James; BAKER, Ryan S.; HUDSON, Scott E. Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. In: *Proceedings of Graphics Interface 2005*. Victoria, British Columbia: Canadian Human-Computer Communications Society, 2005, s. 129–136. GI '05. ISBN 1568812655. Dostupné tiež z: https://dl.acm.org/doi/10.5555/1089508.1089530.

37. Heart Disease Dataset. *Kaggle* [online]. 2019 [cit. 2024-04-25]. Dostupné z : `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`.

38. FANAEE-T, Hadi. *Bike Sharing*. UCI Machine Learning Repository, 2013. Dostupné z DOI: `10.24432/C5W894`.

39. WOLBERG, William; MANGASARIAN, Olvi; STREET, Nick; STREET, W. *Breast Cancer Wisconsin (Diagnostic)*. UCI Machine Learning Repository, 1995. Dostupné z DOI: `10.24432/C5DW2B`.

Obsah príloh