ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta biomedicínského inženýrství
Katedra biomedicínské techniky

Metody pro klasifikaci dat pacientů s podezřením na nádor žaludku

Methods for biomedical data classification of patients with suspect stomach tumor

Bakalářská práce

Studijní program:	Biomedicínská a klinická technika
Studijní obor:	Biomedicínský technik
Vedoucí práce:	Ing. Jakub Novák

Anzhelika Kamkina

Kladno, květen 2016

České vysoké učení technické v Praze, Fakulta biomedicínského inženýrství

Katedra biomedicínské techniky

Akademický rok: 2015/2016

Zadání bakalářské práce

Student:	**Anzhelika Kamkina**
Obor:	Biomedicínský technik
Téma:	**Metody pro klasifikaci dat pacientů s podezřením na nádor žaludku**
Téma anglicky:	Methods for biomedical data classification of patients with suspect stomach tumor

Zásady pro vypracování:

Analyzujte naměřená data o pacientech, u kterých bylo diagnostikováno onemocnění nádoru žaludku na základě stanovení biomarkerů. Vyberte a implementujte několik klasifikačních modelů, které umožní klasifikovat pacienty na základě jejich atributů na pacienty s potvrzeným nádorem žaludku a pacienty, kde se nádor žaludku nepotvrdil. Modely validujte a zhodnoťte dosažené výsledky v podobě kontingenční tabulky, ROC křivky a též vypočtených charakteristik jako senzitivity, specificity, pozitivní a negativní prediktivní hodnoty.

Seznam odborné literatury:

[1] Witten I, Frank Eibe, Hall Mark A, Data Mining: Practical Machine Learning Tools and Techniques, ed. 3rd, 2011, Morgan Kaufmann, 978-0123748560
[2] Pyle Dorian, Data preparation for data mining, ed. 1st, 1999, Morgan Kaufmann Publishers, 978-1558605299
[3] Tesfaye S1, Boulton AJ, Dyck PJ, Freeman R, Horowitz M, Kempler P, Lauria G, Malik RA, Spallone V, Vinik A, Bernardi L, Valensi P; Toronto Diabetic Neuropathy Expert Group, Diabetic neuropathies: update on definitions, diagnostic criteria, estimation of severity, and treatments, Online, Diabetes Care [online], ed. 2010, [Revidováno 12/2010], [Citováno 2015-09-06], ročník 33, číslo 10, DOI: 10.2337/dc10-1303
[4] Hastie Trevor, Tibshirani Robert, Friedman Jerome, The elements of statistical learning: data mining, inference, and prediction, ed. 3rd, 2001, Springer, 978-0387952840

zadání platné do:	30.09.2017
Vedoucí:	Ing. Jakub Novák
Konzultant:	doc. RNDr. Ladislav Pecen, CSc., Ústav informatiky AV ČR

.....
vedoucí katedry / pracoviště

.....
děkan

V Kladně dne 22.02.2016

Prohlášení

Prohlašuji, že jsem bakalářskou práci s názvem Metody pro klasifikaci dat pacientů s podezřením na nádor žaludku vypracovala samostatně a použila k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k závěrečné zprávě.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Kladně dne 19. května 2016

.....
podpis

Poděkování

Děkuji svému vedoucímu Ing. Jakubu Novákovi za pomoc při vedení bakalářské práce. Mé poděkování patří též doc. RNDr. Ladislavu Pecnovi, CSc. za spolupráci při získávání údajů pro výzkumnou část práce a užitečné rady při zpracování. Dále bych ráda poděkovala MUDr. Davidu Šmídovi za poskytnutí dat z FN Plzeň.

Abstrakt

Bakalářská práce se zabývá klasifikací dat pacientů s podezřením na nádor žaludku. Cílem práce bylo navrhnout a vytvořit klasifikační model, který by klasifikoval pacienty na základě jejích změřených parametrů na nemocné s karcinomem žaludku a zdravé jedince. Použity byly dva algoritmy selekce příznaků (SFS a ReliefF) a pět různých klasifikačních modelů (k-NN, SVM, naivní Bayesův klasifikátor, rozhodovací strom a neuronová síť). Výsledky klasifikačních algoritmů jsou porovnány a neuronová síť je vyhodnocena jako nejlepší metoda.

Klíčová slova: klasifikace, karcinom žaludku, klasifikační modely, algoritmy strojového učení

Abstract

This bachelor's thesis deals with the classification of patients with the suspected gastric cancer. The aim was to design and create a classification model that classifies patients based on their measured parameters into patients with gastric cancer and healthy individuals. Two algorithms were used for feature selection (SFS and ReliefF) and five different classification models (k-NN, SVM, naive Bayes classifier, decision tree and neural network) were implemented. The results of classification algorithms are compared and the best method of patients' classification is evaluated. The best method is a neural network.

Keywords: classification, gastric cancer, classification models, machine learning algorithms

Obsah

Seznam zkratek

CV	Křížová validace (Cross-validation)
DT	Rozhodovací strom (Desicion Tree)
***k*-NN**	Metoda k nejbližších sousedů (k-Nearest Neighbour)
LOOCV	Vynech-jednu-instanci křížová validace (Leave-one-out Cross-Validation)
mRMR	minimum-Redundancy-Maximum-Relevance
NBK	Naivní Bayesův klasifikátor (Naive Bayes Classifier)
NBTree	Naive Bayes/Decision-Tree Hybrid
NPV	Negativní prediktivní hodnota (Negative Predictive Value)
PPV	Pozitivní prediktivní hodnota (Positive Predictive Value)
ROC	Receiver Operating Characteristic
SBS	Zpětná eliminace (Sequential Backward Selection)
SFS	Dopředný výběr (Sequential Forward Selection)
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Over-sampling TEchnique
SVM	Metoda podpůrných vektorů (Support Vector Machine)

Úvod

Problém rakoviny zůstává hlavní prioritou pro moderní společnost. Maligní nádory žaludku jsou jednou z nejčastějších příčin úmrtí na zhoubné nádory. Každoročně je diagnostikováno téměř 800 tisíc nových případů a 628 tisíc úmrtí způsobených tímto onemocněním [1]. Karcinom žaludku patří mezi pět nejčastěji vznikajících maligních nádorových onemocnění jak u mužů, tak i u žen [1].

Stanovení správné diagnózy je prvotním a nejdůležitějším krokem pro zahájení kvalitní léčby. Obecně platí, že s progresí nemoci se zvyšuje riziko mortality pacienta, proto je detekce jejích počátečních projevů velmi důležitá [2].

Klasifikace biomedicínských dat hraje významnou roli v predikci a diagnostice chorob. S rozvojem nových informačních technologií je možné použit metody strojového učení pro klasifikaci typu nemoci, určení nejvhodnějšího způsobu léčby, posouzení rizika komplikací [3].

Cílem celé práce je analýza dat pacientů s nádorem žaludku, dále navržení, vytvoření a porovnání několika klasifikačních modelů, které umožní rozdělovat pacienty na základě jejich změřených dat na pacienty s rakovinou žaludku a zdravé jedince. Práce nepředpokládá vytvoření úplně nové klasifikační metody, ale návrh, implementace a nastavení parametrů vhodného klasifikačního modelu z již existujících nástrojů strojového učení.

1 Využití klasifikátorů v medicíně

V současné době je používáni klasifikátorů v medicíně stále více rozšířené. Klasifikátory se využívají pro podporu rozhodování, pro stanovení diagnózy, terapii a prognózy pacienta. Na klasifikaci dat pacientů s metastázami lymfatických uzlin při rakovině žaludku se zaměřuje publikace [4]. Pro klasifikaci dat byl použit klasifikátor k-NN spolu s různými metodami pro výběr atributů a byla dosažena úspěšnost klasifikace 96,3 %. O použití klasifikačního algoritmu k-NN se lze dočíst také v článku [5]. V této publikaci pro klasifikaci chronických jaterních onemocnění se používají dále algoritmy strojového učení SVM a naivní Bayesův klasifikátor. Pro model naivního Bayesova klasifikátoru byl v článku [5] navržen způsob jak zlepšit jeho úspěšnost pomocí boostingu a s jeho pomocí bylo dosaženo lepších výsledků při klasifikaci onemocnění jater. Ke stejné problematice se vyjadřuje práce [6]. V této práci je pro klasifikaci dat použit NBK, rozhodovací strom (DT) a kombinace těchto dvou algoritmů, která se nazývá NBTree. Nejlepší úspěšnost klasifikace byla dosažena při použití metody NBTree. V práci [7] se používá umělá neuronová síť (NN) pro predikci diagnózy viru hepatitidy. V práci je zkoumána úspěšnost neuronové sítě při učení s učitelem a bez učitele a byly vyzkoušeny různé konfigurace architektury neuronové sítě. Lepších výsledků bylo dosaženo při učení s učitelem. Dopředný výběr (SFS) jako algoritmus pro výběr atributů je použit v práci [4].

Pro vyhodnocení výsledků klasifikace se používá přesnost (angl. accuracy)[4] a matice záměn [8].

2 Teoretický základ

Pro vytvoření účinného klasifikačního nástroje je nutné seznámení s problematikou klasifikace biomedicínských dat, získání teoretických předpokladů a znalostí pro jejich následnou aplikaci v praktické části práce.

2.1 Nádor žaludku

Obecně platí, že karcinomem žaludku trpí přibližně dvakrát více mužů než žen a nárůst výskytu nemoci lze pozorovat se zvyšujícím se věkem. Maximální výskyt nemoci v populaci je po sedmdesátém roce života, výskyt nemoci u pacientů do 30 let je zcela vzácný [2]. Metastázy se vyskytují u 80-90 % jedinců s rakovinou žaludku [9]. Dlouhodobě platí, že většina onemocnění je diagnostikována ve 4. stádiu nemoci, což významně zhoršuje prognózu nemocných. Prognóza přežití pacientů s maligním nádorem žaludku je nejvíce ovlivněna stadiem pokročilosti nemoci [2].

50-70 % pacientů s diagnostikovanou rakovinou žaludku v I. stadiu dosahuje pětiletého přežití, ve II. stadiu 50-60 %, ve III. stadiu 20-30 % a ve IV. stadiu méně než 5 % pacientů [10].

Mezi rizikové faktory ovlivňující vznik nádoru žaludku a jeho maligní transformaci patří dietetické návyky, genetická predispozice, kouření a zvýšená konzumace soli, uzených potravin a alkoholu. V řadě studií je prokázána závislost mezi infekcí Helicobacter Pylori a chronickým postižením žaludeční sliznice [11]. Na podkladě chronického zánětu sliznice může dojít k přestavbě buněk, což lze považovat za předstupeň při vzniku karcinomu žaludku. Helicobacter Pylori je dnes uznávaný jako karcinogen I třídy [11].

Velkým problémem u karcinomu žaludku je dlouhodobě bezpříznakový průběh nemoci, což je často příčinou pozdní diagnostiky [9].

Časná stadia onemocnění karcinomu žaludku probíhají bez nápadných klinických znaků a první příznaky bývají natolik nespecifické, že jsou zpravidla člověkem podceněny. Nejčastějšími příznaky tak mohou být pocit plnosti, únava, váhový úbytek [2].

Včasná diagnostika je důležitá pro zahájení léčby a podstoupení chirurgického výkonu. Úspěšná léčba je možná pouze chirurgicky, zejména v raných stadiích onemocnění. Bez chirurgického zákroku je léčba vždy paliativní [2].

2.2 Využití strojového učení

Strojové učení je rozsáhlým podoborem umělé inteligence, který se zabývá způsoby a technikami tvorby algoritmů, které se dokáží učit a přizpůsobovat se změnám. Cílem algoritmů strojového učení je odhalit souvislosti, ukryté ve vstupních datech a na jejich základě určit výstupní údaje. Metody strojového učení lze rozdělit na učení s učitelem a učení bez učitele. Algoritmy učení s učitelem jsou natrénovány na datech, u nichž je znám správný výsledek. Při učení s učitelem je každý vzorek reprezentován dvojící, která se skládá ze vstupního objektu (typicky vektor hodnot) a požadované výstupní hodnoty. Při učení bez učitele nemá algoritmus předem definované výstupní hodnoty a určí je sám [12].

Klasifikace je jedna z hlavních úloh strojového učení. Cílem klasifikace je zařadit objekty do disjunktních tříd. Objekty jsou popsané hodnotami jednotlivých parametrů, které jsou vstupem do klasifikačního algoritmu neboli klasifikátoru [12].

V práci jsou výhradně použity algoritmy, které jako způsob učení používají učení s učitelem, jelikož je předem známo, do které třídy patří vstupní objekt.

2.3 Biomedicínská data a jejich předzpracování

Biomedicínská data mohou být výsledky měření fyziologických parametrů, hodnoty získané z vyšetření pacienta nebo vyhodnocená data ankety. Pro použití dat klasifikačním algoritmem je potřeba je vhodným způsobem upravit. Biomedicínská data jsou velmi nesourodá a nestrukturovaná [13]. Nejprve musejí být odstraněny všechny duplikátní záznamy o pacientech a změřené atributy nesoucí stejnou informaci. Vhodné je provést sjednocení formátu všech záznamů v tabulce. Jistým omezením některých algoritmů strojového učení může být to, že jsou náchylné na odlehlé hodnoty, proto se provádí jejich detekce a odstranění. Dalším důležitým krokem je minimalizovat množství chybějících hodnot buď jejich doplněním nebo kompletním odstraněním atributu nebo pozorování. Data pro klasifikaci nesmí obsahovat atributy, které neovlivňují rozhodování, např. datum pořízení vzorku nebo index instance přiřazený během vyšetření [13].

2.4 Vyvážení skupin

Mezi další možné nedostatky reálných biomedicínských dat patří malé množství vstupních vzorků a také jejich nerovnoměrné rozdělení do tříd [13]. Lze buď

odstranit nadbytečné vzorky u majoritní třídy nebo uměle vytvořit chybějící počet vzorků, které by spadaly do minoritní třídy. Jednou z nejčastěji používaných metod pro vyvážení skupin je algoritmus Synthetic Minority Over-sampling TEchnique (SMOTE).

Metoda SMOTE slouží pro doplnění třídy s menším počtem vzorků pomocí vytvoření syntetických vzorků. Metoda vykazuje větší úspěšnost, než doplňování náhodným opakováním existujících vzorků [14].

Ze skupiny, kterou je nutné doplnit, je vybrán vektor hodnot jednoho vzorku. V celé množině dat se najde k nejbližších sousedů a náhodně se vybere jeden z nich. Vypočte se rozdíl mezi vektorem hodnot původního případu a vektorem hodnot zvoleného nejbližšího souseda. Tento rozdíl je vynásoben náhodným číslem v rozsahu mezi 0 a 1 a přičte se k původnímu vektoru hodnot. To způsobí, že hodnota atributu se bude vždy nacházet v rozsahu mezi původní hodnotou atributu a hodnotou atributu zvoleného souseda. Počet k nejbližších sousedů se volí na základě doplňovaných vzorků [14].

2.5 Normalizace dat

Atributy mohou být měřeny v různých rozsazích. Normalizace je převedení hodnot atributů na stanovený rozsah, jako je například 0 až 1. Normalizace je užitečná pro klasifikaci pomocí algoritmů zahrnujících neuronové sítě nebo měření vzdáleností, jako je metoda k nejbližších sousedů a shlukování. Při použití neuronové sítě se zpětným šířením chyby pro klasifikaci (viz podkapitola 2.7.2), pomůže normalizace vstupních hodnot pro každý atribut zrychlit fázi učení. U metod založených na měření vzdálenosti (viz podkapitola 2.7.1), pomáhá normalizace zabránit přidělení vetší váhy atributům s velkým rozsahem oproti atributům s menším rozsahem hodnot. Mezi nejčastější metody normalizace dat patří: z-score normalizace, min-max normalizace a softmax normalizace [15].

2.5.1 Z-score normalizace

Z-score normalizace je standardizace dat jejich směrodatnou odchylkou. Normalizované hodnoty mají průměr rovný 0 a směrodatnou odchylku 1. Normalizaci metodou z-score lze popsat jako:

$$x_{norm} = \frac{x_{ik} - \bar{x_k}}{\sigma_k}, \tag{1}$$

kde x_{norm} je normalizovaná hodnota, x_{ik} je aktuální hodnota atributu, $\bar{x_k}$ je střední hodnota atributu, σ_k je směrodatná odchylka atributu [15].

2.5.2 Min-Max normalizace

Min-max normalizace neboli standardizace rozpětím je druh lineární transformace dat. Algoritmus normalizuje hodnoty na základě maximální a minimální hodnoty daného atributu. Normalizované hodnoty atributů se tak nacházejí v intervalu 0 až 1. Matematicky lze min-max normalizaci vyjádřit následujícím způsobem:

$$x_{norm} = \frac{x_i - X_{min}}{X_{max} - X_{min}}, \tag{2}$$

kde x_{norm} je normalizovaná hodnota, x_i je aktuální hodnota atributu i, X_{max} a X_{min} jsou maximální a minimální hodnoty atributu [15].

2.5.3 Softmax normalizace

Sigmoidální nebo softmax normalizace je způsob nelineární transformace umožňující snížit vliv extrémních nebo odlehlých hodnot bez jejich odstranění z datového souboru. Normalizaci pomocí logistické sigmoidální funkce lze matematicky vyjádřit jako:

$$x_{norm} = \frac{1}{1 + e^{-\frac{x_i - \mu_k}{\sigma_k}}}, \tag{3}$$

kde x_{norm} je normalizovaná hodnota, x_i je aktuální hodnota atributu i, μ_k je střední hodnota atributu, σ_k je směrodatná odchylka atributu [16].

Logistická sigmoidální funkce omezuje rozsah normalizovaných dat na hodnoty mezi 0 a 1.

2.6 Výběr příznaků

Biomedicínská data často obsahují až desítky tisíc pozorování (počet pacientů) a desítky až stovky změřených atributů [17]. Pro klasifikaci není potřeba používat všechny naměřené atributy. Nadbytečné vstupy zpomalují výpočet, což není žádoucí pro aplikace, které zpracovávají data v reálném čase, a mohou také způsobovat nepřesnost. Velký počet použitých atributů přímo vyžaduje velký počet nastavovaných parametrů klasifikátoru. Složitý model může být přeučený na trénovacích datech a nebude úspěšně klasifikovat na základě dat testovacích. Proto je vhodné vybrat jenom ty příznaky, které jsou maximálně důležité pro správné rozlišení vzorů. Jak

správně určit množinu příznaků je jeden z největších problémů algoritmů strojového učení. Existuje celá řada metod pro zjištění důležitosti atributů, např.: dopředný výběr neboli algoritmus sekvenční dopředné selekce (SFS), zpětná eliminace (SBS), metoda ReliefF nebo lze využít i statistické testy [3].

2.6.1 Dopředný výběr (SFS)

Dopředný výběr je postup, kterým se vybere podmnožina co nejvýznamnějších atributů z celé množiny vstupů. Princip zjištění důležitosti atributů klasifikačním algoritmem spočívá v tom, že se vezme jeden příznak (jeden sloupec v tabulce) a ve všech vzorech se jeho hodnoty nahradí průměrem hodnot daného příznaku. Takto sestavená množina je předložená klasifikátoru, a klasifikátor je ohodnocen z hlediska chyby. Celý postup se opakuje pro všechny atributy v tabulce. Ten příznak, kterému bude odpovídat největší chyba ze všech iterací klasifikátoru, bude nejvíc významný pro zařazení vzorku do správné třídy během klasifikace. Příznak se nadále ponechává jako vybraný a celý postup se opakuje pro zbývající příznaky. Po proběhnutí celého algoritmu lze seřadit atributy podle velikosti chyb klasifikátoru [18].

2.6.2 Metoda ReliefF

Algoritmy z rodiny Relief jsou založené na hledání nejbližších sousedů a výpočtu vzdálenosti mezi pozorováními. Původní algoritmus Relief je schopen pracovat pouze s daty pro binární klasifikaci a je velmi citlivý na zašuměná data. Jeho modifikovanou verzí je algoritmus ReliefF, který umožňuje provádět výběr příznaků pro víc než dvě třídy a není tolik citlivý na chybějící a odlehlé hodnoty [19].

Algoritmus 1 ReliefF

Vstup: M instancí x_k (N atributů a C tříd); Pravděpodobnost třídy p_y; Parametr opakování m; Počet n nejbližších instancí z každé třídy

Výstup: pro každý atribut F_i váha $-1 \leq W_i \leq 1$

```
for i = 1 to N do
    W_i = 0,0;
end for;
for l = 1 to m do
    náhodně vyber instanci x_k (z třídy y_k);
    for y = 1 to C do
        najdi n nejbližších instancí x[j,y] z třídy y, j = 1..n;
        for i = 1 to N do
            for j = 1 to n do
                if y = y_k{nearest hit} then
                    W_i = W_i − diff(i, x_k, x[j,y])/(m · n);
                else
                    W_i = W_i + p_y/(1 − p_yk)·diff(i, x_k, x[j,y])/(m · n);
                end if;
            end for;{j}
        end for;{i}
    end for;{y}
end for;{l}
return(W)
```

Pro dvě instance, které nemají chybějící hodnoty, je funkce *diff* definována jako:

$$diff(i, x_a, x_b) = \frac{\mid x_a(i) - x_b(i) \mid}{\max(x_i) - \min(x_{i)})} \tag{4}$$

Algoritmus náhodně vybírá jednu instanci a pro ní hledá nejbližší vzorek spadající do stejné třídy (angl. *nearest hit*) a nejbližší vzorek, který byl klasifikován do opačné třídy (angl. *nearest miss*). Vektor W_i se aktualizuje v závislosti na hodnotě instance R_i, *nearest hit* a *nearest miss*. Celý proces se opakuje m-krát.

Stanovení počtu atributů se provádí na základě určení konkrétního prahu. Pokud váha atributu překročí tento práh, atribut se vybere. V opačném případě se pro klasifikaci nepoužije. Pokud atribut získal zápornou váhu, znamená to, že je irelevantní pro klasifikaci [19].

2.7 Algoritmy strojového učení

V rámci práce bylo vybráno 5 konkrétních klasifikátorů. Jedná se o k-NN, naivní Bayesův klasifikátor, SVM, rozhodovací strom a neuronovou síť.

2.7.1 Metoda k nejbližších sousedů (k-NN)

Algoritmus je založen na principu hledání nejbližších sousedů. Metoda k nejbližších sousedů (k-NN) je neparametrická metoda pro klasifikaci dat. Jedná se klasifikátor pro učení s učitelem. Jeho princip je umístit dotazovaný prvek do prostoru, kde se nachází data, a najít k nejbližších sousedů. Objekt je pak klasifikován do té třídy, kam patří většina z těchto nejbližších sousedů. Počítá se vzdálenost mezi vstupním vektorem a k vektory z trénovací množiny. Existuje několik metrik pro výpočet vzdálenosti mezi prvky, např. hammingova a euklidovská. Hammingova metrika využívá součet absolutních hodnot, euklidovská metrika je založena na odmocnině ze součtu čtverců vzdálenosti. Počet nejbližších sousedů k se obvykle volí jako liché číslo, aby se zabránilo situacím, kdy stejný počet nejbližších sousedů spadá do dvou nebo více tříd [20].

Použití algoritmů k-NN může být spojeno s několika problémy. Za prvé, výskyt irelevantních proměnných (např. číslo vzorku) ve vstupních datech snižuje přesnost klasifikace, za druhé algoritmus pracuje především s použitím numerických proměnných, kategoriální proměnné mohou být zpracovány, ale musí být speciálně upravené pro algoritmus [20].

2.7.2 Neuronová síť (NN)

Umělá neuronová síť (NN) je matematický model, založený na principu organizace a fungování biologických neuronových sítí – sítí nervových buněk živého organismu. Z hlediska strojového učení, je neuronová síť využívaná pro řešení úloh klasifikace, diskriminační analýzy, shlukování, rozpoznávaní obrazů atd. Neuronová síť je systém propojených jednoduchých procesorů (umělých neuronů) umožnujících interakci. Každý neuron má libovolný počet vstupů, ale pouze jeden výstup. Neurony jsou uspořádané v pravidelných vrstvách. Neuronová síť se většinou skládá ze třech typů vrstev: vstupní, skryté a výstupní. Vstupní vrstva slouží pro hodnoty vstupních proměnných. Každý ze skrytých a výstupních neuronů je spojen se všemi prvky z předchozí vrstvy. Topologie neuronové sítě pro klasifikaci se vyznačuje tím, že počet neuronů ve výstupní vrstvě je obvykle roven počtu definovaných tříd klasi-

fikace. Je-li síti předložen určitý obraz, na jednom z jejích výstupů by se měl objevit příznak toho, že obraz patří do této třídy. Zároveň na dalších výstupech by mělo být vidět, že obraz nepatří do této třídy. Pokud dva nebo více výstupů obsahují znak příslušnosti ke stejné třídě, znamená to, že síť si „není jistá“ se svou odpovědí. Pro úlohu klasifikace se většinou používá perceptronová architektura sítě [16].

2.7.3 Support Vector Machine (SVM)

SVM neboli Support Vector Machine je metoda klasifikace pomocí podpůrných vektorů, která hledá nadrovinu, která by v prostoru příznaků optimálně rozdělila data. Ve své základní formě se jedná o binární klasifikátor, tzn. lze ho použit pouze pro data, rozdělené do dvou tříd. Optimální nadrovinou se rozumí taková rovina, kde všechny body jednotlivých tříd leží v opačných poloprostorech a vzdálenost nejbližších bodů obou tříd od nadroviny je co největší. Jinak řečeno, okolo nadroviny je na obě strany co nejširší pruh bez bodů. Na popis nadroviny stačí pouze nejbližší body, které se nazývají podpůrné vektory (angl. support vectors)[21].

2.7.4 Naivní Bayesův klasifikátor (NBK)

Základním předpokladem naivního Bayesova klasifikátoru je to, že vztahy mezi atributy lze popsat rozložením pravděpodobnosti. NBK používá odhad podmíněné pravděpodobnosti, se kterou vzorek patří do určité třídy. Konkrétně se používá Bayesova věta pro výpočet podmíněné pravděpodobnosti každé hodnoty atributů dané instance a výsledné třídy [22].

Vzhledem k tomu, že tato metoda je založena na zjednodušeném a poněkud nereálném předpokladu, že příznaky jsou podmíněně nezávislé, je tato metoda dobře známá jako naivní Bayesův klasifikátor. Použití NBK je zejména vhodné při velkém počtu vstupních dat. I přes svou jednoduchost je NBK často úspěšnější klasifikátor než sofistikovanější metody klasifikace [22].

2.7.5 Rozhodovací strom (DT)

Rozhodovací strom je jedna z metod strojového učení, určená pro klasifikaci a predikci. Rozhodovací strom je algoritmus rekurzivně dělící množinu vstupních hodnot na dílčí podmnožiny. DT se skládá z větví, uzlů a listů. Každý uzel (místo rozvětvení) stromu reprezentuje podmínku rozhodování podle konkrétního atributu a z uzlu vede konečný počet hran (větví), které představují alternativy rozhodování.

Obvykle se v uzlu porovnává hodnota atributu s konstantní hodnotou z celého oboru hodnot atributů, která je algoritmem zvolena jako rozhodovací práh. Nicméně, některé stromy porovnávají dva atributy mezi sebou. U rozhodovacího stromu rozděluje každý vnitřní uzel trénovací množinu vzorků do dvou nebo více dílčích množin podle převládání vzorků určité třídy v těchto množinách. Koncové uzly (listy) udávají klasifikaci do příslušné třídy, která se vztahuje na všechny případy, které dosáhnou koncového uzlu [23].

2.8 Hodnocení úspěšnosti klasifikace

Po natrénování klasifikátoru následuje jeho testování a hodnocení úspěšnosti klasifikace. Použití natrénovaného klasifikátoru na trénovací data není vhodné pro získání objektivní představy o chování modelu při klasifikaci dosud neznámých dat. Většinou se data rozdělí na trénovací a testovací množinu. Klasifikátor se naučí na trénovací množině dat a je pak následně otestován na testovací množině dat, která nebyla použita při jeho trénování. V případě velkého množství dostupných dat se ještě vytváří tzv. validační množina, která slouží k doladění parametrů klasifikátoru. Také validační množina má za účel zabránit přeučení klasifikátoru tzn. stavu, kdy je systém příliš přizpůsobený množině trénovacích dat a není schopen generalizovat získané znalosti a závislosti. Existuje řada metod, kterými lze zajistit disjunktnost trénovací a testovací množiny pro každou iteraci klasifikátoru, např. hold-out validace, k-násobná křížová validace a leave-one-out validace [3].

2.8.1 Hold-out validace

Nejjednodušším způsobem pro rozdělení dat na trénovací a testovací množinu je náhodné rozdělení v předem stanoveném poměru neboli metoda hold-out. Nejčastěji bývá zvolen poměr 2:1, tzn. že dvě třetiny dat jsou použity pro trénování klasifikátoru a zbytek pro testování. Celý proces trénování a testování je opakovaný několikrát. Pro vyloučení situace, kdy trénovací množina obsahuje pouze instance jedné třídy, lze použit specifickou variantu stratifikované hold-out validace.V trénovací a testovací množině bude stejný poměr zastoupení jednotlivých tříd, jak je tomu u celé množiny dat. Výsledná chyba klasifikace je průměrem všech chyb klasifikace [3].

Výhodou této metody je rychlá a jednoduchá implementace, nevýhodou však je překrývání různých trénovacích a testovacích množin během iterací klasifikace. Tento problém je plně vyřešen metodou k-násobné křížové validace [3].

2.8.2 *k*-násobná křížová validace

Často používaným způsobem pro hodnocení úspěšnosti klasifikace je použití k-násobné křížové validace (angl. cross-validation). Data se náhodně rozdělí do k podmnožin. Každá podmnožina je postupně použita jako testovací množina a klasifikátor se natrénuje na zbývajících $k-1$ množinách dat. Zaznamenává se chyba klasifikace. Celý postup se opakuje celkem k-krát na různých trénovacích a testovacích množinách. Chyby klasifikace se zprůměrují, čímž se získá celkový odhad chyby. Ve výsledku je každý vzorek použit pouze jednou pro trénování a jednou pro testování klasifikátoru. Standardním postupem je provedení k-násobné křížové validace k-krát [3].

Vynech-jednu-instanci (Leave-one-out neboli LOOCV) křížová validace je limitním případem křížové validace. To je n-násobná křížová validace, kde n je počet vzorků v datasetu. Každý vzorek je postupně vynecháván pro testování klasifikátoru (tzn. pro zařazení do třídy), klasifikátor se natrénuje na hodnotách všech zbývajících vzorků, jejichž počet se rovná $n-1$. Leave-one-out křížová validace nevyžaduje opakování celého procesu, protože výsledek klasifikace bude vždy stejný [3].

Vzhledem k tomu, že křížová validace nepoužívá všechna data pro vytvoření modelu, tak je běžně používanou metodou pro zabránění přeučení klasifikátoru v průběhu učení. Křížová validace je vhodná v případě malého množství vstupních dat, avšak její nevýhodou může být velká výpočetní náročnost [3].

2.8.3 Matice záměn

Matice záměn je forma kontingenční tabulky znázorňující rozdíly mezi skutečnými a předpokládanými třídami pro sadu označených vzorků. Sloupce jsou předpokládané třídy a řádky jsou skutečné třídy. V matici záměn TP („true positive“) je počet skutečně pozitivních výsledků (tzn. kolik nemocných bylo správně diagnostikováno jako nemocný), FP („false positive“) je počet negativních výsledků, nesprávně klasifikovaných jako pozitivní (tzn. kolik zdravých lidí bylo chybně diagnostikováno jako nemocný), jako TN („true negative“) je označován počet správně klasifikovaných negativních výsledků (tzn. kolik zdravých lidí bylo správně diagnostikováno jako zdraví), FN („false negative“) je počet pozitivních výsledků, nesprávně klasifikovaných jako negativní (tzn. kolik nemocných bylo chybně diagnostikováno jako zdraví) [12].

Tabulka 1: Matice záměn

	Predikovaný pozitivní	Predikovaný negativní
Skutečně pozitivní	TP	FN
Skutečně negativní	FP	TN

Senzitivita popisuje schopnost testu správně detekovat pacienty, kteří mají nemoc. Senzitivita testu je pravděpodobnost, se kterou test zachytí případ onemocnění u skutečně nemocného pacientu. Senzitivita testu může být matematicky vyjádřena jako:

$$Senzitivita = \frac{TP}{TP + FN}. \tag{5}$$

Specificita popisuje schopnost testu správně detekovat zdravé jedince. Specificita je podíl správně klasifikovaných subjektů, u kterých není přítomné sledované onemocnění, z celé kontrolní skupiny. Specificitu testu lze vyjádřit jako:

$$Specificita = \frac{TN}{TN + FP}. \tag{6}$$

Pozitivní prediktivní hodnota (PPV) je pravděpodobnost přítomnosti nemoci při pozitivním výsledku testu. Pozitivní prediktivní hodnotu lze matematicky popsat jako:

$$PPV = \frac{TP}{TP + FP}. \tag{7}$$

Negativní prediktivní hodnota (NPV) vyjadřuje pravděpodobnost nepřítomnosti nemoci při negativním výsledků testu. Negativní prediktivní hodnota:

$$NPV = \frac{TN}{TN + FN}. \tag{8}$$

Celková správnost testu neboli úspěšnost vyjadřuje podíl správně klasifikovaných pacientů ze všech testovacích subjektů:

$$Úspěšnost = \frac{TP + TN}{TP + TN + FP + FN}. \tag{9}$$

2.8.4 ROC křivka a AUC plocha

ROC křivka je účinný nástroj pro hodnocení úspěšnosti klasifikátoru. ROC křivka graficky znázorňuje výsledky binární klasifikace algoritmů strojového učení. Na svislé ose se vynáší relativní četnost skutečně pozitivních vzorů vyjádřená jako procentní podíl z celkového počtu pozitivních, což je pravděpodobnost, že jako správný bude

vyhodnocen pozitivní případ. Na vodorovné ose je znázorněna relativní četnost skutečně negativních případů, vyjádřená jako procentní podíl z celkového počtu negativních případů. Každý bod ROC charakteristiky je dán dvěma hodnotami – FPR ($1 - senzitivita$) a TPR ($specificita$). Nastavováním různých prahových hodnot se na ROC křivce hledá kompromis mezi počtem falešně pozitivních a falešně negativních případů, jinak řečeno mezi senzitivitou a specificitou [24].

Kvantitativní interpretaci ROC křivky dává AUC (angl. *area under curve* – plocha pod křivkou) – oblast ohraničená ROC křivkou a osou podílu falešně pozitivních klasifikací. Hodnota AUC udává pravděpodobnost správného klasifikování jedince, který byl vybrán ze skupiny pacientů a z kontrolní skupiny. Čím vyšší je hodnota AUC , tím lepší je klasifikátor. AUC může nabývat hodnot od 0 do 1. Hodnota 1 odpovídá ideálnímu klasifikátoru, který má 100 % senzitivitu, hodnota 0,5 ukazuje nevhodnost zvoleného způsobu třídění (úspěšnost klasifikace odpovídá náhodnému odhadu) [25].

3 Teoretická část

Celý proces od analýzy dat do navržení klasifikátorů byl rozdělen do těchto částí:

- Předzpracování dat
- Výběr nejdůležitějších atributů
- Hledání vhodných parametrů nastavení každého klasifikátoru
- Trénování klasifikátorů
- Testování klasifikátorů
- Porovnání výsledků klasifikace

3.1 Reálná vstupní data

Vstupními daty pro praktickou část práce je tabulka ve formátu xlsx, která obsahuje data pacientů z FN v Plzni. Data poskytl MUDr. David Šmíd. Tabulka obsahovala 79 záznamů o 23 atributech.

Tabulka shrnuje informace o osobních údajích pacientů, o hodnotách parametrů, změřených během vyšetření a výsledné skupině, do které patří pacient. Všechny parametry potřebné pro stanovení diagnózy byly získané ze vzorku krve pacienta během laboratorního vyšetření. Prediktivními atributy jsou tzv. nádorové markery, což jsou laboratorně prokazatelné známky, kterými je určité nádorové onemocnění charakteristické. Diagnóza nádoru žaludku byla potvrzena biopsií.

Atributy byly jak kvantitativní, diskrétní a spojité veličiny (např. věk pacienta, množství protilátek k Helicobacter Pylori ve vzorku), tak kategoriální (např. skupina, do které spadá pacient).

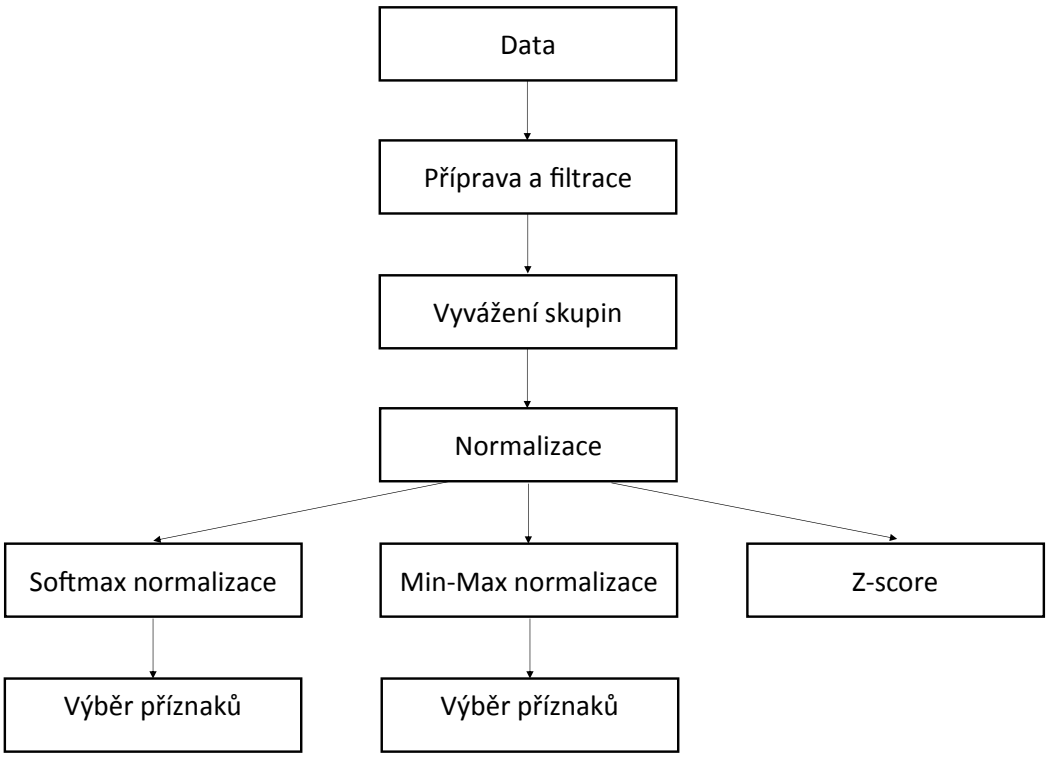
Vstupní dataset byl nevyvážený, což znamená že nemá rovnoměrné zastoupení jednotlivých tříd. 36 pacientů spadá do skupiny „nemocný“, což znamená že má prokázaný karcinom žaludku, 43 pacientů spadá do skupiny s názvem „kontrolní skupina“, což vypovídá o tom, že na základě změřených parametrů se nádor žaludku nepotvrdil.

3.2 Předzpracování dat

Pro samotnou klasifikaci je nutné předzpracování dat, které se skládá z přípravy dat a jejich filtrace. Cílem předzpracování dat je vybrat pouze ty údaje, které jsou

relevantní pro klasifikaci a prezentovat je ve tvaru, vhodném pro zpracování klasifikačním algoritmem.

Na Obrázku 1 je znázorněn průběh předzpracování dat pro následnou klasifikaci.



Obrázek 1: Postup předzpracování dat pro klasifikaci

3.2.1 Příprava dat

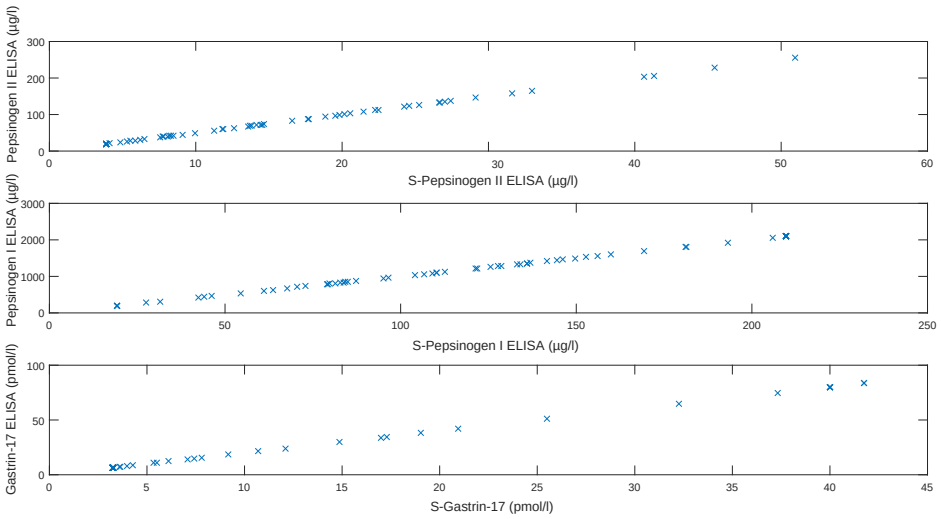
Prvním krokem přípravy dat bylo odstranění barevných značení v tabulce a sjednocení záznamů. Dál byly odstraněny názvy jednotlivých atributů a pořadové číslo, které bylo přiřazeno pacientovi v průběhu analýzy. Slovní značení pacienta bylo nahrazeno číslem. Ženské pohlaví bylo označeno jako 0, mužské pohlaví jako 1.

Slovní značení skupiny, do které spadá pacient, bylo nahrazeno číslem. Pacienti, kteří měli potvrzenou diagnózu nádoru žaludku byli zařazeni do skupiny 1, kontrolní skupina pacientů byla označena jako skupina 0. Sloupec, který obsahuje cílový atribut, byl extrahován do samostatné tabulky.

3.2.2 Filtrace dat

Prvním krokem bylo vyřešit problém duplicity dat. 12 pacientů absolvovalo vyšetření dvakrát. Byly vytvořeny dvě tabulky, které obsahovaly záznamy pouze z prvního a pouze z druhého vyšetření. V důsledku toho, že hodnoty změřené při prvním a při druhém vyšetření byly podobné a občas neměnné, byla v dalších krocích předzpracování využita pouze jedna z tabulek.

Dále byla stanovena míra vzájemné korelace atributů pomocí Pearsonova korelačního koeficientu. Bylo zjištěno, že tři páry atributů mají hodnotu korelačního koeficientu rovnou 1, což svědčí o přítomnosti přímé lineární závislosti. Tři vstupní atributy byly stanoveny dvěma různými laboratorními metodami. Na obrázku 2 je zobrazena vzájemná korelace třech párů vstupních atributů. Každý graf ukazuje vzájemnou lineární závislost dané dvojice atributů.



Obrázek 2: Tři dvojice korelovaných vstupních atributů

Použití všech šesti atributů pro klasifikaci je zbytečné, proto je potřeba odstranit ty atributy, které méně korelují s výslednou skupinou pacienta. Bylo zjištěno, že všech šest atributů má stejnou míru závislosti s výslednou skupinou, proto byly odstraněny atributy, které jsou násobené koeficientem. To znamená, že ze šesti atributů byly vybrány tři: S-Pepsinogen II ELISA, S-Pepsinogen I ELISA, S-Gastrin-17.

Výsledná upravená tabulka obsahuje data pro 67 pacientů, u kterých je změřených 18 parametrů. Dataset obsahuje 38 záznamů pacientů mužského pohlaví a 31 záznamů pacientů ženského pohlaví. Do třídy 1 spadá 24 pacientů, kteří mají karcinom žaludku. Do třídy 0 je zařazeno 43 pacientů, kteří nemají karcinom žaludku

(tzn. jsou zdraví).

3.3 Vyvážení skupin

Nerovnoměrné zastoupení tříd, do kterých probíhá klasifikace, může ovlivnit její úspěšnost, proto bylo dalším důležitým krokem při předpracování dat vyvážení skupin.

Třída 1 obsahovala o 19 vzorků méně než třída 0. Pro zachování důležitých rysů vstupních dat nebyly odstraněny přebytečné vzorky z třídy 0, ale minoritní třída 1 byla doplněna 19 vzorky metodou SMOTE.

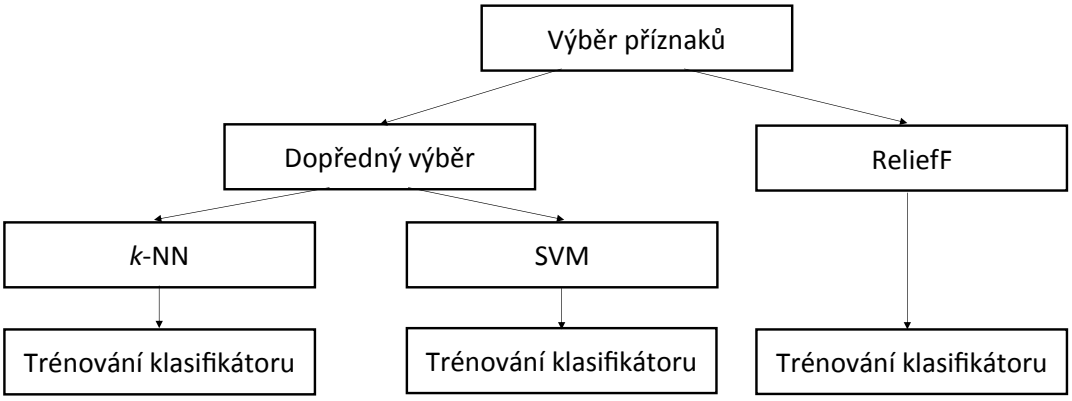
Při implementaci metody SMOTE byl použit pseudokód převzatý z [14].

3.4 Normalizace dat

Byly zvoleny tři metody normalizace dat: z-score normalizace, softmax normalizace i min-max normalizace. Metoda z-score normalizace byla implementována pomocí funkce *zscore*, jejíž vstupem je matice vstupních atributů, výstupem je matice normalizovaných hodnot, kde každý sloupec má střední hodnotu 0 a směrodatnou odchylku 1. Metody min-max a softmax notmalizace byly implementovány podle vzorců (2) a (3).Všechny hodnoty atributů po softmax a min-max normalizaci se nacházely v intervalu $\langle 0; 1 \rangle$.

3.5 Důležitost atributů

Následujícím krokem bylo zjištění důležitosti atributů a výběr jejich optimálního počtu postupně pomocí metody dopředného výběru a metody ReliefF. Na Obrázku 6 je vidět schéma průběhu fáze výběru příznaků.



Obrázek 3: Schéma výběru příznaků

3.5.1 Dopředný výběr

Algoritmus dopředného výběru atributů byl implementován pomocí klasifikátorů k-NN a SVM. Vstupem pro algoritmy byla změřená data pacientů s jedním atributem vždy nahrazeným průměrem jeho hodnot. Algoritmus byl natrénován na takto vzniklé podmnožině dat a zaznamenala se chyba klasifikace. Postupně byly nahrazeny všechny atributy. Ten atribut, kterému ze všech iterací odpovídala největší chyba klasifikace, byl považován za důležitý a ponechával se jako vybraný. Celý postup byl opakován pro všechny atributy, výsledkem byla posloupnost atributů, seřazených podle jejich důležitosti pro správné zatřídění vzorků.

Klasifikátor k-NN byl trénován pomocí funkce *fitcknn* (viz podkapitola 3.7.1). Klasifikátor SVM byl natrénován pomocí funkce *fitcsvm* (viz podkapitola 3.7.3). Chyba klasifikace pro obě metody byla vypočítána pomocí funkce *loss*. Vstupem pro funkci *loss* byl natrénovaný klasifikátor, vstupní a cílové atributy. Výstupem byl podíl nesprávných klasifikací.

3.5.2 Meroda ReliefF

Metoda ReliefF byla implementována pomocí funkce *relieff*. Vstupem pro ní byly změřené atributy pacientů, jejích výsledná třída a počet nejbližších instancí k. Výstupem funkce byla posloupnost indexů atributů seřazených podle jejích důležitosti a jejích váhy.

Seřazení atributů a velikost jejich vah jsou závislé na počtu nejbližších sousedů. Pro klasifikační úlohy se z každé třídy vybírá k nejbližších sousedů. Je-li počet k

nastaven na 1, algoritmus může být náchylný na zašuměná data. Blíží-li se počet nejbližších sousedů k k počtu případů v datech, tak algoritmus nemusí být schopen najít důležité atributy. Proto podle článku [19] je doporučený počet k rovný 10.

3.6 Výběr příznaků

Z posloupnosti seřazených atributů je potřeba vybrat jenom ty, které významně ovlivňují úspěšnost a jsou maximálně důležité pro správné zatřídění vzorů. Metoda SFS byla implementována pomocí dvou klasifikátorů: SVM a k-NN, které měly stejné nastavení parametrů jako při zjištění důležitosti atributů.

3.6.1 Dopředný výběr (SFS)

Pro výběr určitého množství atributů z posloupnosti seřazených atributů je potřeba stanovit kritérium. Pomocí metod k-NN a SVM byl implementován následující algoritmus. Jako vstup do klasifikátoru se použije první nejdůležitější atribut z posloupnosti dříve seřazených atributů, vypočte se chyba klasifikátoru, poté se k němu přidá druhý nejdůležitější atribut atd. Postupně tak bude prozkoumán příspěvek všech atributů na chybu klasifikace. Celý postup se opakuje dokud se zmenšuje chyba klasifikace. Klasifikátory k-NN a SVM byly postupně trénovány na stále se zvětšující množině atributů, seřazených podle jejich důležitosti. Atributy byly přidávané po jednom a zaznamenávala se chyba klasifikace. Výsledkem algoritmu byla seřazená posloupnost atributů. V ideálním případě se zvolí množství atributů, při jejichž použití bylo dosažené nulové chyby klasifikátoru. Pokud chyba klasifikace v průběhu dopředného výběru nenabývá nulových hodnot, zvolí se množství atributů, kdy je chyba klasifikace minimální či již neroste.

3.6.2 Metoda ReliefF

Po seřazení atributů podle důležitosti následuje redukce celé množiny vstupních atributů na menší podmnožiny. Obecně platí, že kladnější váhu mají atributy, které jsou důležitější pro klasifikaci. Záporná váha atributu vypovídá o jeho irelevantnosti pro klasifikaci. Množinu atributů lze vybrat buď stanovením nějakého prahu pro hodnoty váhy nebo použitím pouze atributů s kladnou váhou pokud práh nelze jednoznačně stanovit.

3.6.3 Nejčastěji vybírané atributy

Pro zjištění nejčastěji vybíraných atributů byl vytvořen následující algoritmus, jehož pseudokód je uveden v Algoritmu 2.

Algoritmus 2 Nejčastěji vybírané atributy

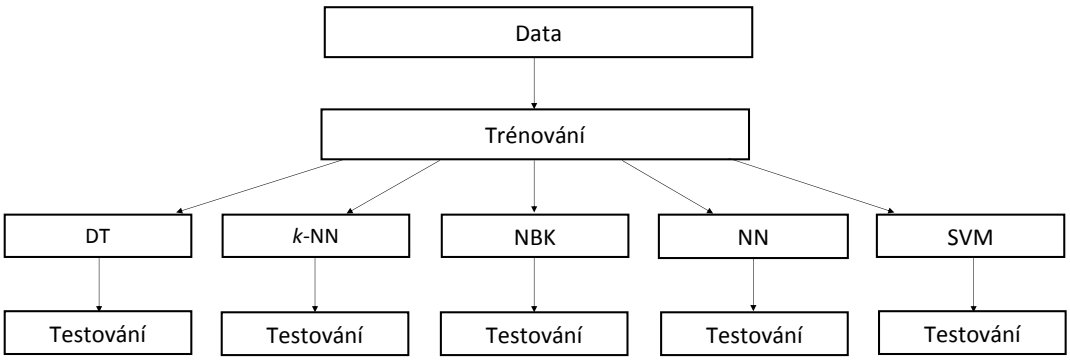
Vstup: n podmnožin vybraných atributů, vektor vah atributů W, počet atributů k

Výstup: pro každý atribut x_i váha W_i

```
for j = 1 to n do
  for i = 1 to k do
    if atribut x_i je přítomen v podmnožině j then
      W_i = W_xi ;
    else
      W_i = 0;
    end if;
    W(j, i) = W_i
  end for;{i}
end for;{j}
W = W/n; return(W)
```

3.7 Algoritmy strojového učení

Bylo vybráno a implementováno 5 klasifikačních algoritmů: k-NN, neuronová síť, SVM, NBK a DT. Zvolené algoritmy jsou založeny na principu učení s učitelem. Všechny algoritmy prováděly klasifikaci do dvou tříd. Data byla před trénováním vždycky rozdělena na trénovací a testovací množinu. Nastavení a natrénování klasifikátoru probíhalo na trénovací množině, pro testování úspěšnosti se používala testovací množina dat. Na Obrázku 4 je zobrazeno schéma trénování jednotlivých klasifikátorů.



Obrázek 4: Schéma trénování klasifikátorů

3.7.1 Metoda *k*-NN

Metoda k-NN byla implementována pomocí funkce *fitcknn*, jejíž vstupem jsou vstupní atributy a cílové atributy, výstupem je natrénovaný klasifikační model. Pro výpočet vzdálenosti mezi prvky byla zvolena euklidovská metrika. Počet k nejbližších sousedů byl iterativně zvolen na základě úspěšnosti klasifikace (viz podkapitola 4.5.1). Při klasifikaci byl neznámý vektor klasifikován do třídy, do které spadá nejvíc nejbližších sousedů.

3.7.2 Neuronová síť

Pro klasifikaci dat byla vytvořená dopředná perceptronová neuronová síť pomocí funkce *patternnet*. Neuronová síť se skládala ze třech vrstev: vstupní, jedné skryté a výstupní. Množství vstupů bylo závislé na počtu vstupních atributů, výstupní vrstva se skládala ze dvou výstupů podle počtu tříd, do kterých bude probíhat klasifikace. Ve skryté vrstvě se používá sigmoidální přenosová funkce, ve výstupní vrstvě softmax přenosová funkce.

Vytvořená neuronová síť byla natrénována pomocí funkce *train*. Výstupem funkce je natrénovaná neuronová síť a množství provedených epoch učení. Trénovacím algoritmem byla zvolena metoda konjugovaného gradientu z důvodu její mírné výpočtové a časové náročnosti a dostatečné poskytované přesnosti při rozpoznávaní obrazů. Metoda konjugovaného gradientu je modifikovaným algoritmem zpětného šíření chyby. Používá standardní numerické optimalizační metody. Pro natrénování neuronové sítě je třeba přizpůsobit synaptické váhy a prahy každého neuronu takovým způsobem, aby se snižovala chyba mezi výstupním a požadovaným signálem. Chyba se počítá pomocí chybové funkce vzájemné entropie a cílem sítě je upravit

matice vah a prahů tak, aby byla dosažena minimální chyba.

Pro trénování neuronové sítě bylo potřeba upravit tabulku, poskytující cílový atribut pro klasifikaci. Byla vytvořena matice o 2 řádcích a 86 sloupcích, kde počet sloupců je počet pacientů, každý řádek reprezentuje jednu třídu klasifikace. Příslušnost k určité třídě se označuje hodnotou 1 v řádku, odpovídajícím této třídě, a hodnotou 0 ve všech ostatních řádcích.

3.7.3 Support Vector Machine

Klasifikátor SVM byl vytvořen a natrénován pomocí funkce *fitcsvm*. Jako typ funkce jádra (angl. *kernel*) byla zvolena lineární funkce, která je výchozím nastavením při binární klasifikaci dat a rozděluje data pomocí optimální separační nadroviny. Pro urychlení učícího procesu byla zvolena metoda nalezení optimální nadroviny Sequential Minimal Optimization (SMO), která uchovává v paměti počítače pouze vybranou podmatici z jádra matice [26].

3.7.4 Naivní Bayesův klasifikátor

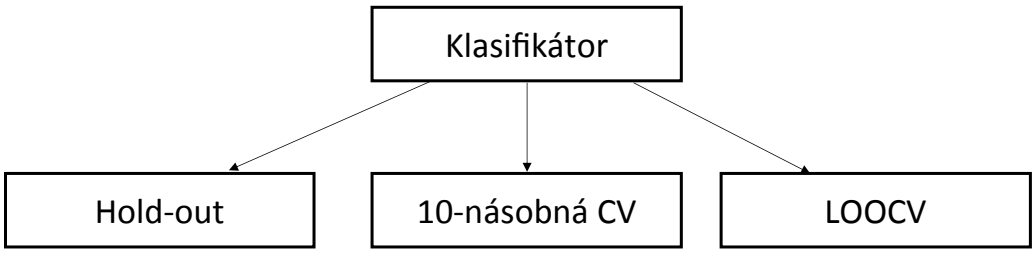
NBK byl natrénován pomocí funkce *fitcnb*. NBK počítá hustotu rozložení pravděpodobnosti dat a klasifikuje objekt do třídy s větší pravděpodobností. Byl využit odhad pravděpodobnosti pomocí gaussovské křivky. Pravděpodobnost zastoupení jednotlivých tříd bylo nastaveno na [0,5 0,5], jak je tomu v reálných datech.

3.7.5 Rozhodovací strom

Byl vytvořen a natrénován rozhodovací klasifikační strom pomocí funkce *fitctree*. Nastavení optimálního počtu úrovní rozhodovacího stromu probíhalo automaticky učením se ze vstupních dat. Strom nebyl prořezáván (tzn. nebyla snižována složitost modelu odstraňováním redundantních atributů a podstromů). Pro sestavení stromu s nejjednodušší architekturou byla zkoumána optimální sekvence rozhodovacích atributů, použitých pro větvení. Maximální počet rozhodovacích uzlů byl nastaven na (*počet případů* − 1). Jako kritérium pro volbu nejlepšího atributu pro dělení byla použita tzv. Giniho míra neboli míra diverzity uzlu. Giniho míra hledá v trénovacích datech největší třídu závislé proměnné a odděluje ji od ostatních dat. Dalšími možnostmi rozhodovacích kriterií je entropie a informační zisk atributů [3].

3.8 Testování klasifikátorů

Na Obrázku 5 je znázorněné schéma testování klasifikátorů. Pro vyhodnocení klasifikační přesnosti byly použity tři metody validace.



Obrázek 5: Schéma testování klasifikátorů

V případě hold-out validace byl celý postup trénování a testování klasifikátorů opakován stokrát. Pro implementaci hold-out validace byla vytvořena funkce *Rozdeleni*. Výstupem funkce byly indexy případů pro trénovací a testovací množinu. Rozdělení dat bylo náhodné v poměru 2:1, tzn. klasifikátor byl natrénován na dvou třetinách vstupních dat (62 instancí) a následně otestován na zbylé jedné třetině dat (28 instancí). Hold-out validace byla stratifikovaná, v trénovací i testovací množině zastoupení jednotlivých tříd bylo v poměru 1:1.

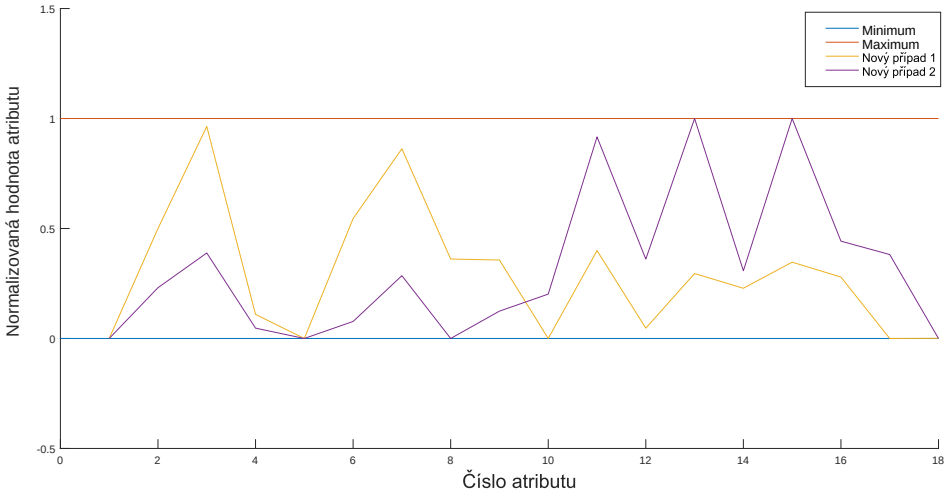
Dále byla pro všechny podmnožiny dat použita desetinásobná křížová validace, všechny algoritmy byly puštěny stokrát a zaznamenávala se chyba klasifikace. Data byla na začátku náhodně rozdělena na 10 přibližně stejně velkých podmnožin, trénování klasifikátorů probíhalo na 9 podmnožinách dat, jedna podmnožina byla vždy vynechaná pro testování. 10-násobná křížová validace byla provedena pomocí funkce *crossval*. Třetí metodou testování klasifikátoru byla metoda LOOCV. Klasifikační algoritmus byl natrénován na 85 instancích, jeden vzorek byl následně použit pro testování klasifikátorů.

4 Praktická část

V praktické části práce bude názorně předvedeno konkretní nastavení klasifikačních algoritmů. Všechny algoritmy byly implementovány v programovém prostředí MATLAB R2015b s použitím Statistics and Machine Learning a Neural Nerwork toolboxů.

4.1 Vyvážení skupin

Byl implementován algoritmus SMOTE pro vygenerování syntetických případů za účelem vyvážení skupin. Počet nových vzorků, které bylo potřeba vygenerovat, tvořil 79 % od původního množství vzorků, proto počet k nejbližších sousedů byl stanoven na 5 [14]. Na Obrázku 6 jsou zobrazené dva náhodně vybrané nové případy generované metodou SMOTE. Je vidět, že hodnota každého atributu se nachází mezi hodnotou vybraného případu a jeho zvoleného souseda.



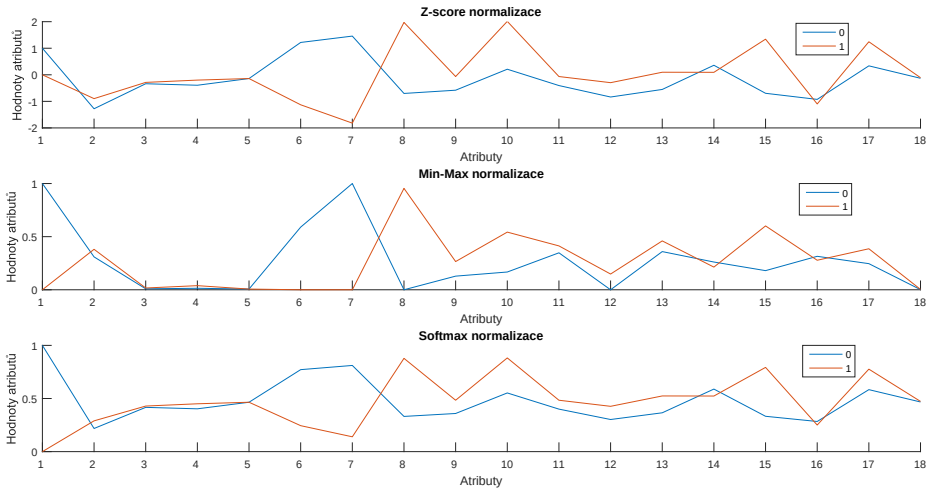
Obrázek 6: Zobrazení dvou nových případů, vygenerovaných metodou SMOTE

Výsledkem vyvážení skupin je tabulka, obsahující 86 vzorků se změřenými 18 příznaky a poměrem zdravých ku nemocným rovným 43:43.

4.2 Normalizace dat

Byly zvoleny a implementovány tři metody normalizace dat: z-score normalizace, min-max normalizace a softmax normalizace. Pro zkoumání úspěšnosti všech tří

metod normalizace byly zobrazené dva náhodně vybrané dva vzorky, které spadají do třídy 0 a 1.



Obrázek 7: Zobrazení normalizovaných hodnot vzorků pomocí metody rovnoběžných souřadnic

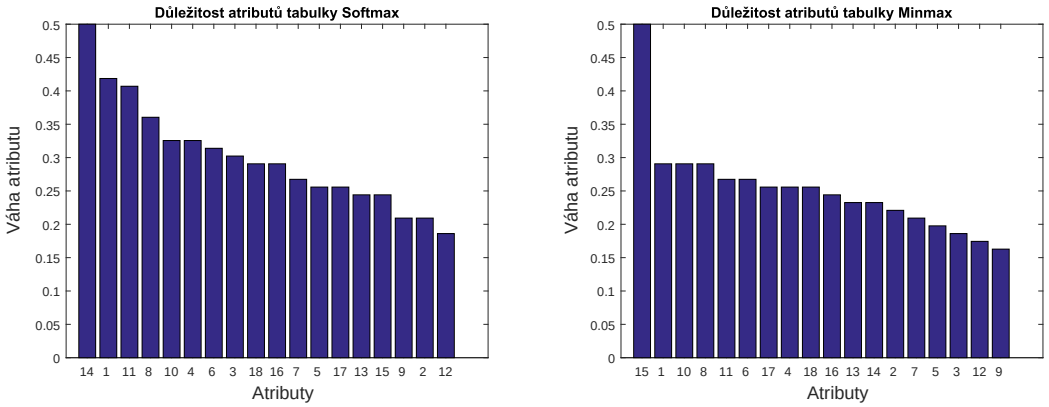
Z Obrázku 7 je patrné, že metoda normalizace z-score nepřevedla data do rozsahu s průměrnou hodnotou rovnou 0 a směrodatnou odchylkou rovnou 1, proto v dalších částech práce nebyla data normalizovaná touto metodou použitá. Vstupem pro algoritmy výběru nejdůležitějších atributů byly dvě tabulky hodnot, jedna normalizovaná pomocí softmax normalizace a druhá pomocí min-max normalizace.

4.3 Důležitost atributů

Byly zvoleny a implementovány tři metody pro zjištění důležitosti atributů: metoda SFS implementovaná pomocí k-NN, metoda SFS implementovaná pomocí SVM a metoda ReliefF.

4.3.1 Metoda *k*-NN

Pomocí metody SFS implementováné pomocí k-NN byla zjištěna důležitost jednotlivých atributů při klasifikaci. Celý postup byl proveden pro tabulku min-max a softmax normalizovaných dat (viz Obrázek 8). Graf zobrazuje vliv přítomnosti daných atributů na úspěšnost klasifikace.

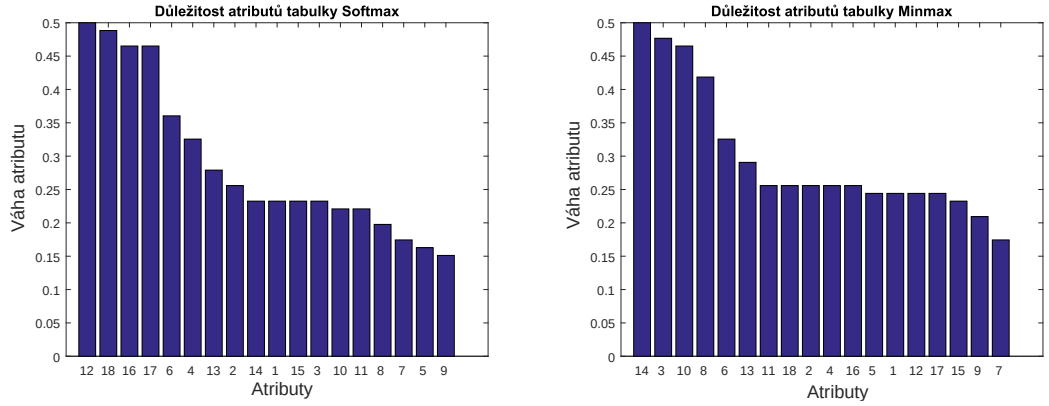


Obrázek 8: Závislost chyby klasifikace na nepřítomnosti daného atributu tabulky softmax normalizovaných a min-max normalizovaných hodnot, zjištěná pomocí metody k-NN

Čím větší váhu má atribut, tím významnější je jeho přítomnost pro správné zatřídění vzorků.

4.3.2 Metoda SVM

Pomocí metody SFS implementované pomocí SVM byla získána posloupnost atributů seřazených podle jejich důležitosti (viz Obrázek 9). Graf zobrazuje vliv přítomnosti daných atributů na úspěšnost klasifikace.

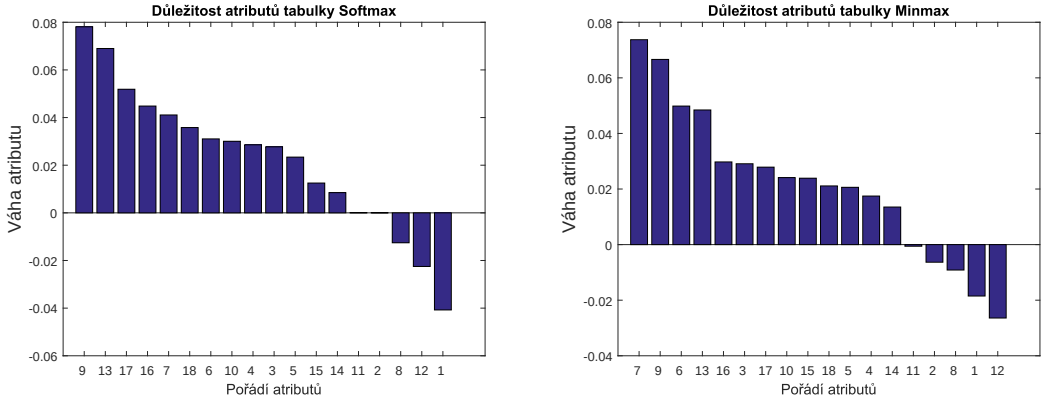


Obrázek 9: Závislost chyby klasifikátoru na nepřítomnosti daného atributu tabulky softmax normalizovaných hodnot a min-max normalizovaných hodnot při klasifikaci, zjištěná pomocí metody SVM

4.3.3 Metoda ReliefF

Třetí metodou zjišťování důležitosti atributů byla zvolena metoda ReliefF. Touto metodou byly atributy seřazené podle ohodnocení jejích důležitosti. Výsledná po-

sloupnost atributů je na Obrázku 10.



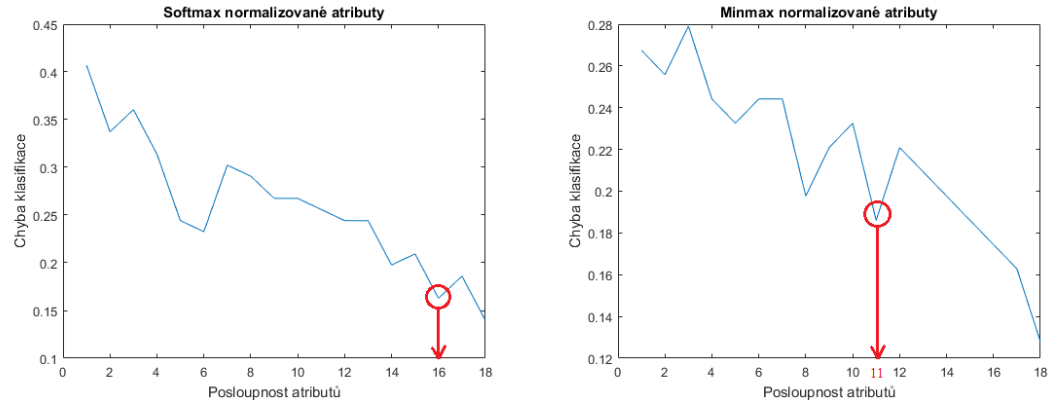
Obrázek 10: Důležitost vstupních atributů tabulky softmax normalizovaných hodnot a min-max normalizovaných hodnot, zjištěná pomocí metody ReliefF

4.4 Výběr příznaků

Z posloupnosti seřazených atributů je potřeba vybrat jenom ty, které významně ovlivňují úspěšnost klasifikace. Hledání nejlepší kombinace vstupních atributů bylo provedeno pomocí metod k-NN a SVM. Vstupem pro algoritmy byla posloupnost atributů, seřazena podle jejich důležitosti (viz podkapitola 4.3.1 a 4.3.2).

4.4.1 Metoda *k*-NN

Prvním algoritmem pro omezení posloupnosti seřazených atributů na podmnožiny byla metoda k-NN. Z grafů na Obrázku 11 je vidět, že chyba klasifikace v závislosti na počtu použitých atributů nikdy nebyla nulová.



Obrázek 11: Závislost klasifikační chyby na počtu atributů tabulek softmax normalizovaných hodnot a min-max normalizovaných hodnot, zjištěná pomocí metody k-NN. Červenou šipkou je označeno množství vybraných atributů.

Pro zjištění optimálního počtu atributů pro klasifikaci byla prozkoumána následná úspěšnost algoritmu k-NN při použití takových množství atributů, kde křivka chyby klasifikace má lokální minimum. Z tabulky 2 je vidět, že největší úspěšnost klasifikace hodnot z tabulky hodnot softmax je při použití prvních 16 atributů ze seřazené posloupnosti. Proto bylo vybráno 16 atributů.

Tabulka 2: Úspěšnost algoritmu k-NN v závislosti na počtu vybraných atributů pro tabulku softmax normalizovaných dat

Počet vybraných atributů	6	9	14	16	18
Úspěšnost (%)	67,02	69,64	79,16	81,19	80,24

Pro tabulku min-max normalizovaných hodnot největší úspěšnost algoritmu výběru atributů byla dosažena při použití prvních 11 atributů seřazených podle důležitosti (viz Tabulka 3). Proto bylo vybráno 11 atributů.

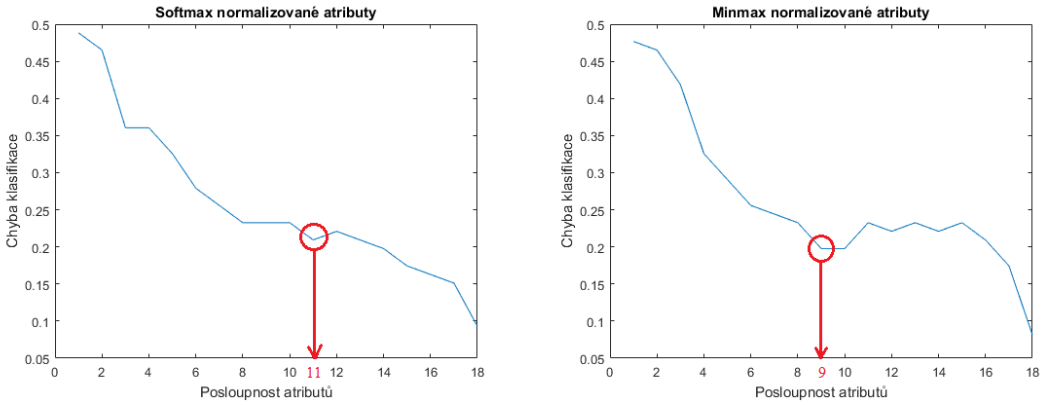
Tabulka 3: Úspěšnost algoritmu k-NN v závislosti na počtu vybraných atributů pro tabulku min-max normalizovaných dat

Počet vybraných atributů	5	8	11	18
Úspěšnost (%)	70,60	73,70	75,12	70,75

4.4.2 Metoda SVM

Dalším zvoleným algoritmem pro výběr atributů byla metoda SVM. Z grafu na Obrázku 12 je patrné, že chyba klasifikace nebyla nikdy nulová. Bylo potřeba vybrat takové množství atributů, při jejichž použití je chyba klasifikace minimální či dlouho neměnná.

Pro tabulku softmax normalizovaných dat bylo zvoleno prvních 11 atributů, protože dle trendu křivky se jedná o bod zlomu, tedy lokální minimum. Pro tabulku min-max normalizovaných hodnot bylo zvoleno prvních 9 atributů, protože se v tomto případě jedná o lokální minimum.



Obrázek 12: Závislost klasifikační chyby na počtu atributů tabulek softmax normalizovaných hodnot a min-max normalizovaných hodnot, zjištěná pomocí metody SVM. Červenou šipkou je označeno množství vybraných atributů.

4.4.3 Metoda ReliefF

Dalším krokem byl výběr atributů, seřazených podle důležitosti algoritmem ReliefF (viz Obrázek 10). Důležitější atributy mají kladnější váhu. Avšak z důvodu přítomnosti atributů se zápornou váhou, které mohou způsobovat chybu klasifikace, bylo rozhodnuto vybrat pouze atributy s kladnou hodnotou váhy. Pro tabulku softmax normalizovaných hodnot bylo zvoleno prvních 15 atributů, pro tabulku min-max normalizovaných hodnot prvních 13 atributů.

4.5 Algoritmy strojového učení

Vstupem do fáze trénování každého klasifikátoru bylo postupně 8 podmnožin dat: 3 podmnožiny dat, získané z tabulky softmax normalizovaných hodnot, 3 podmnožiny dat, získané z tabulky min-max normalizovaných hodnot a kompletní tabulky softmax normalizovaných hodnot a min-max normalizovaných hodnot. Kompletní tabulky softmax a min-max normalizováných hodnot slouží pro zjištění vlivu různých metod redukce dimensionality na úspěšnost klasifikace a jejich porovnání.

4.5.1 Metoda *k*-NN

Vstupními parametry bylo postupně 8 podmnožin dat, množina cílových atributů a koeficient k.

Pro zjištění optimální hodnoty parametru k byla postupně provedena klasifikace s počtem nejbližších sousedů od 5 do 11 s krokem 2. Ideální hodnota k byla vybrána

iterativně na základě úspěšnosti klasifikátoru při daném nastavení k. V tabulkách 4 a 5 je zaznamenána průměrná úspěšnost klasifikace v závislosti na počtu nejbližších sousedů pro jednotlivé podmnožiny dat.

Tabulka 4: Průměrná úspěšnost klasifikace v závislosti na počtu nejbližších sousedů k pro tabulku softmax normalizovaných hodnot

Počet sousedů k	SFS k-NN (%)	SFS SVM (%)	Relieff (%)
5	80,7	76,1	79,9
7	80,3	75,3	77,8
9	80,8	72,9	80,1
11	80,9	71,8	79,7

Pro následující trénování klasifikátoru z důvodu největší úspěšnosti klasifikace při daném počtu k na množině SFS k-NN byl parametr k nastaven na 11, pro množinu SFS SVM k byl nastaven na 5 a pro množinu ReliefF byla zvolena hodnota k rovna 9.

Tabulka 5: Průměrná úspěšnost klasifikace v závislosti na počtu nejbližších sousedů k pro tabulku min-max normalizovaných hodnot

Počet sousedů k	SFS k-NN (%)	SFS SVM (%)	Relieff (%)
5	76,9	76,5	80,0
7	76,1	75,1	78,2
9	75,7	73,6	78,1
11	76,0	75,6	79,3

Pro následující trénování klasifikátoru k-NN na všech podmnožinách min-max normalizovaných hodnot z důvodu největší úspěšnosti klasifikace při daném počtu k byl počet nejbližších sousedů nastaven na 5.

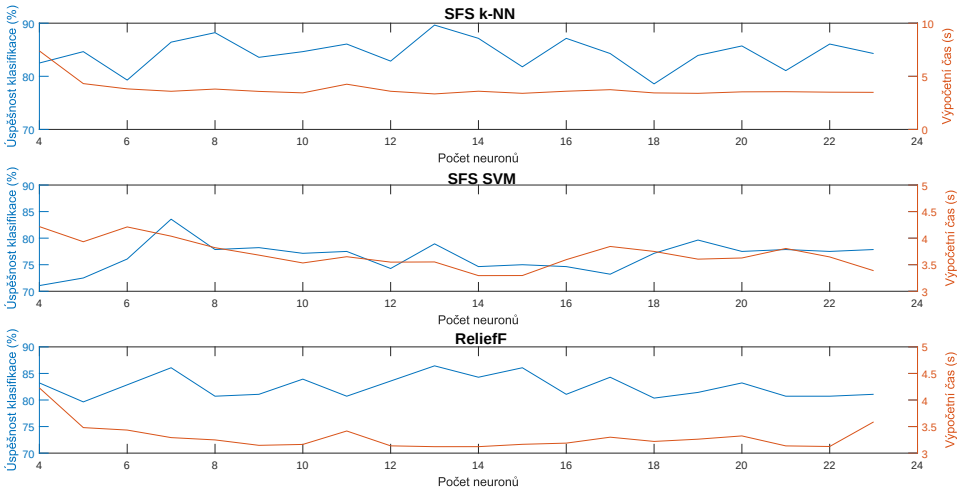
4.5.2 Neuronová síť

Důležitým krokem bylo zjistit potřebný počet neuronů ve skryté vrstvě pro klasifikaci každé podmnožiny dat. Optimální byl počet neuronů, který vedl k největší úspěšnosti. V případě shodných úspěšností bylo třeba najít další parametr ovlivňující algoritmus. Jedním takovým je výpočetní čas.

V případě softmax normalizovaných dat pro podmnožinu dat SFS k-NN bylo rozhodnuto použit 13 neuronů ve skryté vrstvě z důvodu největší dosažené úspěšnosti klasifikace 89,6 % při daném počtu neuronů.

Pro podmnožinu dat, vybraných metodou SFS SVM nejlepší úspěšnost klasifikace 83,6 % byla zaznamenaná při použití 7 neuronů.

Pro podmnožinu dat ReliefF vybraný počet neuronů pro následující trénování klasifikátoru je 13 z důvodu největší úspěšnosti a zároveň nejmenšího výpočetního času (viz Obrázek 13 a 14).

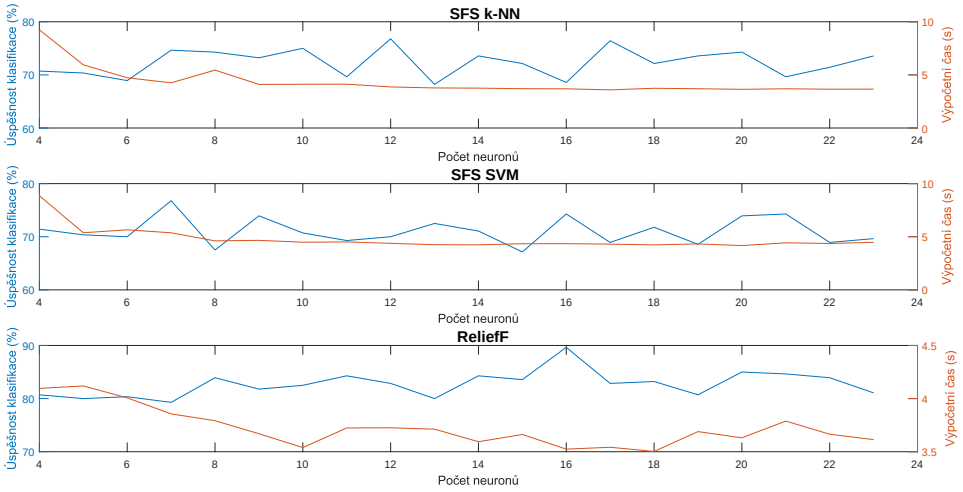


Obrázek 13: Závislost úspěšnosti klasifikace NN na počtu neuronů ve skryté vrstvě, zjištěna pro tabulku softmax normalizovaných hodnot

V případě min-max normalizovaných dat pro podmnožinu dat SFS k-NN bylo zvoleno použit 12 neuronů ve skryté vrstvě z důvodu největší dosažené úspěšnosti 76,8 % při daném počtu neuronů ve skryté vrstvě.

Pro podmnožinu dat SFS SVM nejlepší úspěšnost klasifikace 76,8 % byla zaznamenaná při použití 7 neuronů.

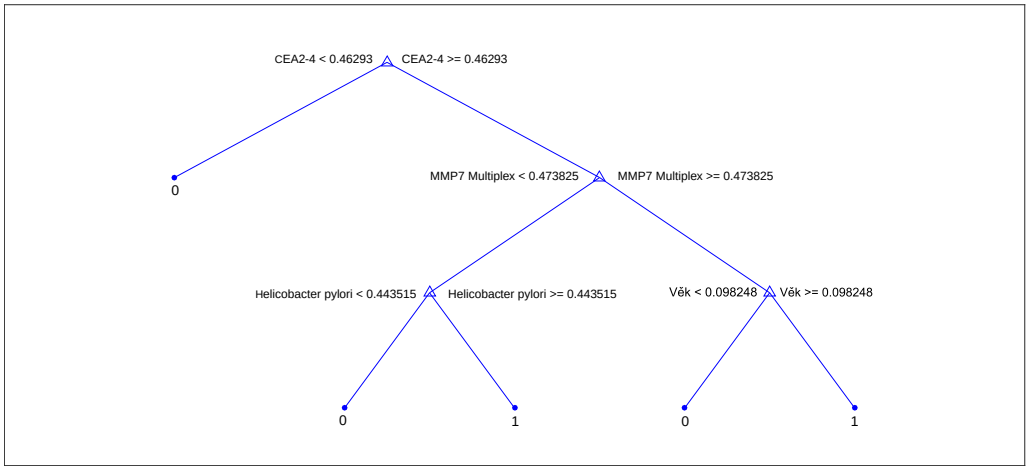
Pro podmnožinu dat ReliefF vybraný počet neuronů pro následující trénování klasifikátoru je 16 z důvodu největší dosažené úspěšnosti klasifikace.



Obrázek 14: Závislost úspěšnosti klasifikace NN na počtu neuronů ve skryté vrstvě, zjištěna pro tabulku min-max normalizovaných hodnot

4.5.3 Rozhodovací strom

Byl vytvořen a natrénován klasifikační rozhodovací strom s použitím nastavení, uvedeném v podkapitole 3.7.5. Pro jednotlivé trénovací podmnožiny byly počet rozhodovacích uzlů, úrovní a složitost struktury rozhodovacích stromů různé. Na Obrázku 15 je zobrazen nejlepší rozhodovací strom, který vykazoval největší úspěšnost a měl nejjednodušší strukturu. Nejlepší rozhodovací strom byl tvořen 4 uzly, ve kterých probíhalo rozhodování. Uzly byly uspořádané ve 3 úrovních.



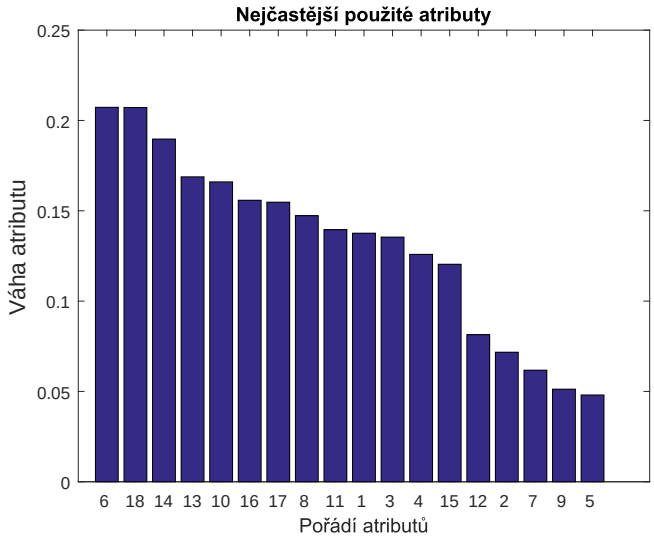
Obrázek 15: Zobrazení struktury nejlepšího rozhodovacího stromu

4.5.4 Zbylé klasifikátory

Klasifikátory SVM a NBK byly natrénovány a otestovány s výchozím nastavením parametrů (viz podkapitola 3.7.3 a 3.7.4).

5 Výsledky

Pěti atributy, které byly nejčastěji zvoleny algoritmy výběru atributů jako nezbytně nutné pro klasifikaci, jsou: S-Pepsinogen II ELISA, PIVKA-II, MMP8 Multiplex, MMP7 Multiplex, MMP1 Multiplex. Atribut S-Pepsinogen II Elisa byl vybrán každým algoritmem z každého datasetu. Atributem, který byl vybraný jako relevantní pro klasifikaci pouze jednou, je CA72-4, který patří mezi nádorové markery.



Obrázek 16: Zobrazení atributů, nejčastěji selektovaných pro klasifikaci

Celkem bylo natrénováno 5 různých klasifikačních algoritmů na 8 různých podmnožinách dat, všechny algoritmy byly validovány třemi způsoby. Pro všechny natrénované klasifikační algoritmy byly sestavené matice záměn. Matice záměn porovnávají u všech tříd vztah mezi referenčními daty a výsledky klasifikace. Každý řádek matice reprezentuje aktuální třídu a každý sloupec reprezentuje třídy predikované pomocí klasifikátorů. Z matice záměn byly vypočítané charakteristiky jako senzitivita, specificita, PPV a NPV.

Z důvodu velkého množství natrénovaných klasifikátorů jsou zobrazené v Tabulce 6 pouze nejlepší výsledky klasifikace z hlediska úspěšnosti, všechny výsledky jsou umístěné v Příloze A a Příloze B. Veškeré výsledky v Tabulce 6 byly dosažené na podmnožinách softmax normalizovaných hodnot a jsou uvedené ve tvaru „průměr ± směrodatná odchylka“.

Tabulka 6: Nejlepší dosažené výsledky klasifikace pro jednotlivé algoritmy

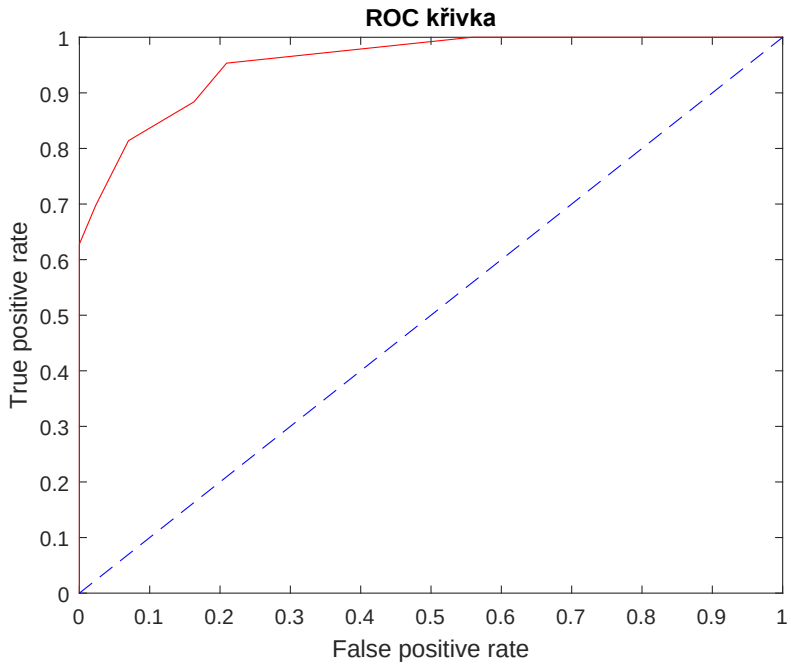
Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS k-NN	74,9 ± 1,8	88,3 ± 2,4	86,6 ± 2,4	77,9 ± 1,3	81,6 ± 1,5
NN	SFS k-NN	85,6 ± 2,4	87,0 ± 2,7	86,8 ± 2,5	85,8 ± 2,1	92,8 ± 2,2
SVM	Celá tabulka	86,0 ± 0,0	88,4 ± 0,0	88,1 ± 0,0	86,4 ± 0,0	87,2 ± 0,0
NBK	ReliefF	65,1 ± 0,0	95,3 ± 0,0	93,3 ± 0,0	73,2 ± 0,0	80,2 ± 0,0
DT	ReliefF	90,7 ± 0,0	90,7 ± 0,0	90,7 ± 0,0	90,7 ± 0,0	90,7 ± 0,0

Nejvhodnějším klasifikátorem je z pohledu úspěšnosti klasifikace umělá neuronová síť. Úspěšnost klasifikace je (92,8 ± 2,2) %. Nejlepší výsledek klasifikace z pohledu úspěšnosti byl dosažen na podmnožině softmax normalizovaných vstupních atributů, vybraných pomocí metody SFS k-NN. Druhého nejlepšího výsledku dosáhl rozhodovací strom, který byl trénován na podmnožině softmax normalizovaných atributů vybrané pomocí algoritmu ReliefF a validován pomocí LOOCV.

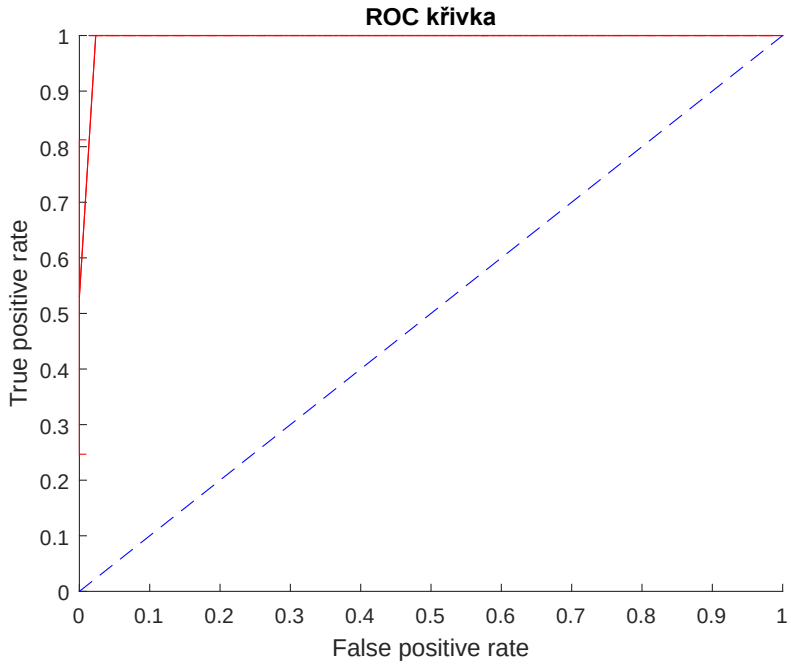
Zároveň byly pro všechny klasifikační algoritmy sestrojené ROC křivky. Z důvodu velkého množství natrénovaných klasifikátorů jsou zobrazené pouze ROC křivky s největší hodnotou AUC pro každý klasifikátor.

V případě cyklického průběhu algoritmu při hold-out a 10-násobné křížové validaci je zobrazena zprůměrovaná ROC křivka spolu s intervalem spolehlivosti, který činí jednu směrodatnou odchylku.

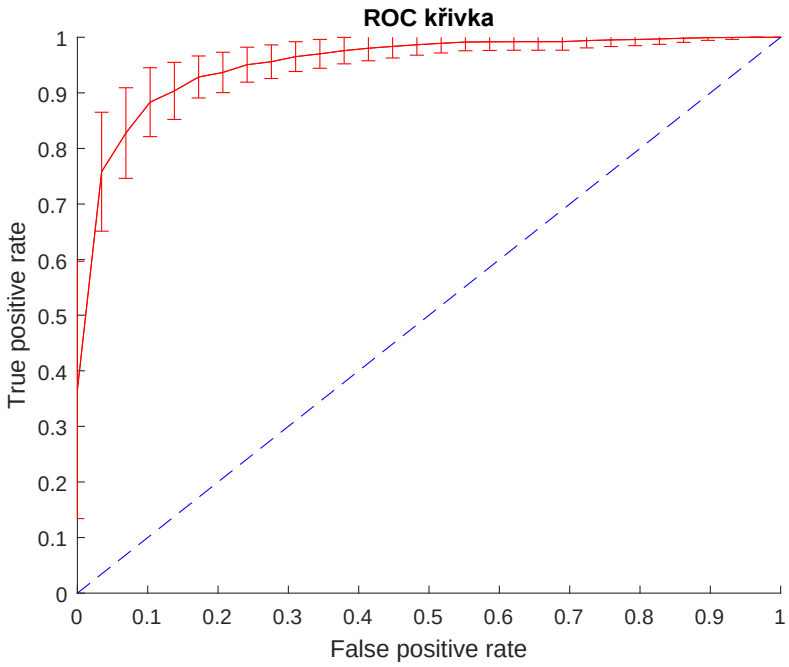
Nejlepším klasifikačním algoritmem z pohledu velikosti AUC je rozhodovací strom natrénovaný na podmnožině softmax normalizovaných dat vybraných pomocí algoritmu ReliefF. Druhou nejvyšší hodnotu plochy pod ROC křivkou vykazovala neuronová síť.



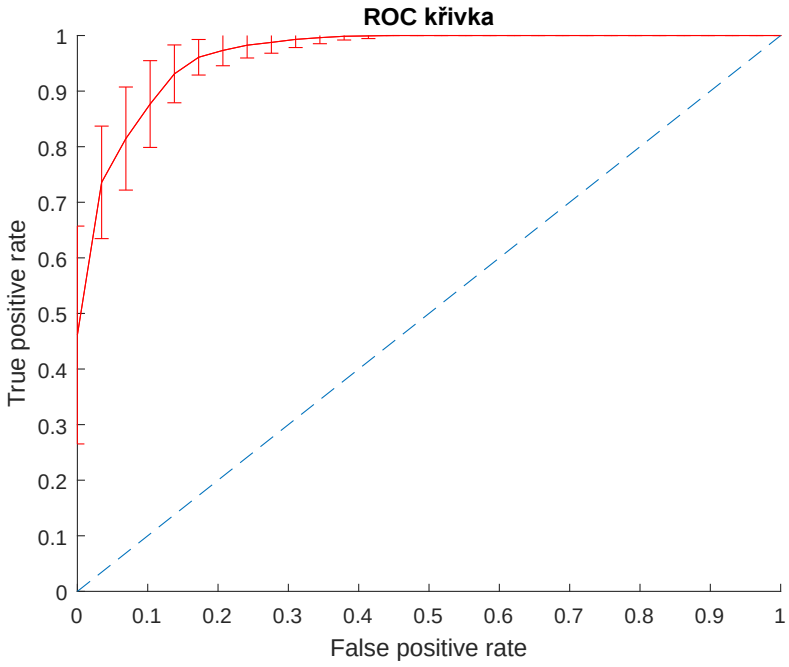
Obrázek 17: Nejlepší ROC křivka pro klasifikátor k-NN (AUC = 0,955)



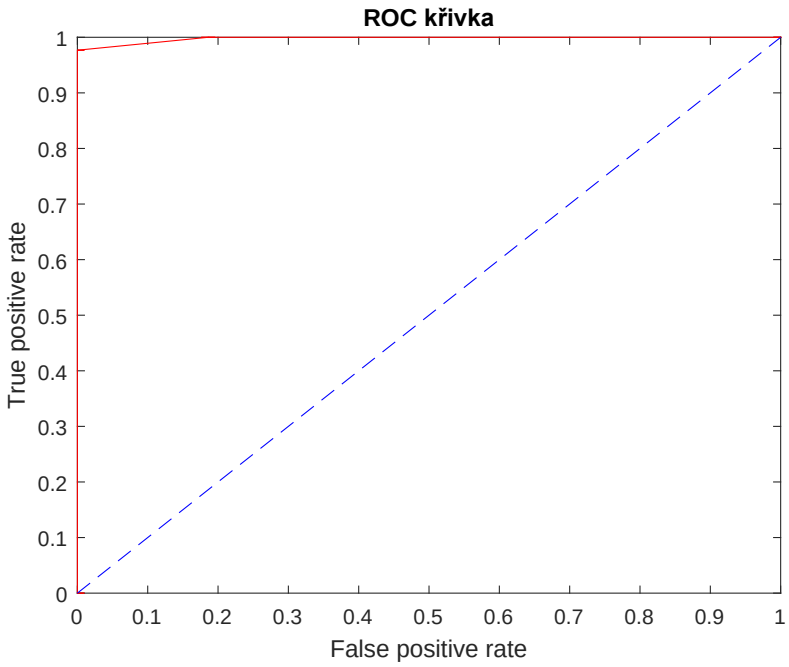
Obrázek 18: Nejlepší ROC křivka pro klasifikátor NN (AUC = 0,995)



Obrázek 19: Nejlepší ROC křivka pro klasifikátor SVM (AUC = 0,962)



Obrázek 20: Nejlepší ROC křivka pro klasifikátor NBK (AUC = 0,973)



Obrázek 21: Nejlepší ROC křivka pro klasifikátor DT (AUC = 0,998)

6 Diskuse

Ve fázi předzpracování dat a jejích přípravy pro následnou klasifikaci byly vyzkoušeny tři metody normalizace dat. Z-score normalizace do rozsahu dat s nulovým průměrem a jednotkovým rozptylem nebyla úspěšně provedena kvůli velkému rozptylu hodnot atributů, proto v další části práce nebyla použita. Všechny nejlepší výsledky klasifikátorů byly dosaženy na podmnožinách softmax normalizovaných atributů. Obecně lze tvrdit, že použití softmax normalizace je úspěšnější než min-max normalizace.

V části důležitost atributů byla zvolená metoda dopředného výběru a metoda ReliefF. Metoda dopředného výběru byla vybrána z důvodu provázanosti s klasifikátory, na které se práce zaměřuje. Metoda ReliefF byla zvolena z důvodu jejího rozšířeného použití a nenáchylnosti na zašuměná data. Lze bylo použit i další metody pro zjištění důležitosti atributů, například metodu mRMR, zpětnou eliminaci nebo statistické testy.

V části výběru příznaků bylo zvoleno použití klasifikátoru SVM a k-NN. Z výsledku je vidět, že oba klasifikátory byly schopny zredukovat počet atributů vstupních dat pro výsledný klasifikátor.

Pro klasifikaci dat byly vybrány následné algoritmy strojového učení: rozhodovací strom, k-NN, naivní Bayesovský klasifikátor a neuronová síť. Výsledky klasifikace byly různé v závislosti na nastavení parametrů algoritmů. Klasifikátor k-NN nejlépe využíval příznaky selektované dopředným výběrem (SFS k-NN). Tyto příznaky byly také nejvhodnější pro neuronovou síť, ačkoli tento algoritmus umí dobře pracovat i s daty vybranými pomocí metody ReliefF. Rozhodovací strom vykazoval největší úspěšnost při použití atributů, vybraných pomocí algoritmu ReliefF nebo při použití kompletní tabulky dat, protože má v sobě již vestavěnou funkci výběru optimální posloupnosti atributů pro rozhodování. Nelze jednoznačně říct, který algoritmus pro výběr atributů byl nejvhodnější pro následnou klasifikaci pomocí NBK. Avšak obecně následné trénování všech klasifikátorů mělo větší úspěšnost na podmnožinách atributů, které byly vybrané pomocí algoritmu ReliefF. Důležitým faktem je, že podmnožina atributů vybrána pomocí SFS SVM neposkytla nejlepší výsledek klasifikace ani pro jeden klasifikátor. Možným důvodem bylo nevhodné zvolení prahu ve fázi výběru atributů.

Úspěšnost jednotlivých klasifikátorů byla určována pomocí hold-out validace, 10-násobné křížové cross-validace a leave-one-out validace. Výsledky získané hold-out

validací výrazně kolísaly, měly velkou směrodatnou odchylku a nebyly uspokojivé. Důvodem k tomu mohlo sloužit to, že množství trénovacích dat nebylo v daném případě dostačující pro úspěšné naučení klasifikátoru. Leave-one-out validace a 10-násobná křížová validace poskytovaly téměř stejné výsledky. Časová náročnost trénování byla menší v případě LOOCV.

Pro vyhodnocení úspěšnosti byla vybrána matice záměn z důvodu její přehlednosti a možnosti dalšího využití pro výpočet jednotlivých parametrů klasifikace jako senzitivita, specificita, PPV, NPV a úspěšnost. Také z důvodu přehledného grafického znázornění závislosti pravdivě pozitivní míry na falešně pozitivní míře v závislosti na měnícím se prahu rozhodování byla použita ROC křivka.

Všechny natrénované klasifikátory NBK neposkytovaly oproti ostatním klasifikátorům vysokou úspěšnost ani senzitivitu, ale měly vysokou specificitu což znamená, že většina lidí bez nádoru žaludku byly klasifikátorem označené jako zdraví.

Lze říct, že oba klasifikační algoritmy DT a NN dosáhly nejlepších výsledků při klasifikaci dat pacientů s podezřením na nádor žaludku. Další možností výzkumu v dané oblasti by bylo nalezení klasifikátoru s nejlepší kombinací velikosti úspěšnosti a plochy pod ROC křivkou. Ten výzkum by mohl být proveden pro větší množství vstupních dat nebo dokonce i pro nová data.

Závěr

Cílem bakalářské práce bylo analyzovat data pacientů s podezřením na nádor žaludku, dále vytvořit klasifikační model, který bude schopen klasifikovat pacienty na základě jejich změřených parametrů na pacienty s nádorem žaludku a zdravé jedince. Práce se zabývá návrhem a vytvářením modelů pro klasifikaci pacientů s použitím algoritmů strojového učení.

Teoretická část práce je zaměřená na návrh průběhu celého experimentu a definování použitých nástrojů a metod. V praktické části byl analyzován datový soubor z Fakultní nemocnice v Plzni. Bylo realizováno 5 klasifikačních modelů: k-NN, SVM, neuronová síť, naivní Bayesův klasifikátor a rozhodovací strom. Celkem bylo natrénováno 120 klasifikátorů (5 klasifikačních modelů, použitých na 8 podmnožinách dat a validovaných třemi způsoby). Klasifikátory jsou schopny rozdělit pacienty na zdravé jedince a nemocné s nádorem žaludku. Nejvhodnějším klasifikátorem pro klasifikaci pacientů do tříd z hlediska velikosti úspěšnosti je umělá neuronová síť. Úspěšnost klasifikace je (92,8 $\pm$ 2,2) %. Nejlepší výsledek klasifikace byl dosažen na podmnožině softmax normalizovaných hodnot, vybrané pomocí metody SFS k-NN. Nejvhodnějším klasifikátorem pro klasifikaci pacientů do tříd z hlediska velikosti AUC je rozhodovací strom. Hodnota AUC je 0,998.

Literatura

[1] Fact sheet 297. *World Health Organization.* [online]. © 2016 [cit. 2016-04-09]. Dostupné z : http://www.who.int/mediacentre/factsheets/fs297/en/

[2] BECKER, Horst D. *Chirurgická onkologie.* Vyd. 1. Praha: Grada, 2005, 854 s. ISBN 80-247-0720-9.

[3] WITTEN, Ian H. a Eibe FRANK. *Data mining: practical machine learning tools and techniques.* 2nd ed. Boston, MA: Morgan Kaufman, 2005. ISBN 0-12-088407-0.

[4] LI, Chao, Shuheng ZHANG, Huan ZHANG, Lifang PANG, Kinman LAM, Chun HUI a Su ZHANG. Using the K-Nearest Neighbor Algorithm for the Classification of Lymph Node Metastasis in Gastric Cancer. In:*Computational and Mathematical Methods in Medicine* [online]. 2012, s.1-11 [cit. 2016-04-09]. Dostupné z: http://www.hindawi.com/journals/cmmm/2012/876545/

[5] RIBEIRO, Ricardo T., Rui T. MARINHO a J. Miguel SANCHES. Classification and Staging of Chronic Liver Disease From Multimodal Data. In: *IEEE Transactions on biomedical engineering* [online]. 2013(5), s.1336-1344 [cit.2016-04-09]. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6387584

[6] ALFISAHRIN, Sadiyah Noor Novita a Teddy MANTORO. Data Mining Techniques for Optimization of Liver Disease Classification. In:*International Conference on Advanced Computer Science Applications and Technologies 2013* [online]. IEEE, 2013, s.379-384 [cit.2016-04-09]. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6836610

[7] ANSARI Sana, SHAFI I., ANSARI A., AHMAD J. a S. I. SHAH. Diagnosis of Liver Disease Induced by Hepatitis Virus using Artificial Neural Networks. In: *2011 IEEE International Multitopic Conference (INMIC)* [online]. s.8-12, 2011 [cit.2016-05-09]. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6151515& tag=1

[8] KHALAF, Mohammed, Abir Jaafar HUSSAIN, Dhiya AL-JUMEILY, Russell KEENAN, Paul FERGUS a Ibrahim Olatunji IDOWU. Robust Approach for Medical Data Classification and Deploying Self-Care Management System for Sickle Cell Disease. In:*2015 IEEE International Conference on Computer and*

Information Technology [online]. IEEE, 2015, s.575-580 [cit.2016-05-09]. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7363123

[9] TOMPKINS, Marilyn B. *Gastric cancer research trends.* New York: Nova Biomedical Books, ©2007. ISBN 978-1600217258.

[10] ADAM, Zdeněk, Jiří VANÍČEK a Jiří VORLÍČEK. *Diagnostické a léčebné postupy u maligních chorob.* 2., aktualiz. a dopl. vyd. Praha: Grada, 2004. ISBN 80-247-0896-5.

[11] HATAKEYMA M., HIGASHI H. Helicobacter pylori CagA: a new paradigm for bacterial carcinogenesis. In: *Cancer Science*, 2005, 96(12). s.835—843. Dostupné z : http://onlinelibrary.wiley.com/doi/10.1111/j.1349-7006.2005.00130.x/pdf

[12] MARSLAND, Stephen. *Machine Learning: An Algorithmic Perspective.* 2nd ed. CRC Press, 2015. ISBN 978-1498759786.

[13] MALLEY, James D. a Sinisa PAJEVIC. *Statistical Learning for Biomedical Data.* Cambridge University Press, 2011. ISBN 978-1139496858.

[14] CHAWLA, Nitesh, Kevin BOWYER, Lawrence HALL a Philip KEGELMEYER. SMOTE: Synthetic Minority Over-sampling Technique. In: *Journal of Artificial Intelligenece Research.* 2002, 16. s.321-357.

[15] JIAWEI, Han a Micheline KAMBER. *Data mining concepts and techniques.* 2nd ed. Amsterdam: Elsevier, 2006. ISBN 978-0-08047-558-5.

[16] PRIDDY, Kevin L. a Paul E. KELLER. *Artificial neural networks: an introduction.* Bellingham, Wash.: SPIE Press, ©2005. ISBN 978-0819459879.

[17] SIDHU, Amandeep S. a Tharam S. DILLON (eds.). *Biomedical data and applications.* Berlin: Springer Verlag, 2009. ISBN 978-3642021923.

[18] JIŘINA, Marcel. *Rozdělení dat do trénovacích a testovacích množin.* In: Biomedicínské inženýrství před námi. Praha: ARSCI, 2007, s.41-46. ISBN 978-80-86078-83-0 (in Czech).

[19] LIU, Huan a Hiroshi MOTODA (eds.). *Computational methods of feature selection.* Boca Raton: Chapman & Hall/CRC, 2008. ISBN 978-1-58488-878-9.

[20] NISBET, Robert a John ELDER. *Handbook of statistical analysis and data mining applications.* Burlington, MA: Academic Press/Elsevier, 2009. ISBN 978-0080912035.

[21] ABE, Shigeo. *Support vector machines for pattern classification.* 2nd ed. New York: Springer, ©2010. ISBN 978-1-84628-838-8.

[22] MAIMON, Oded a Lior ROKACH. *Decomposition methodology for knowledge discovery and data mining: theory and applications.* London: World Scientific, ©2005. ISBN 981-256-079-3.

[23] ROKACH, Lior a Oded MAIMON. *Data Mining with Decision Trees: Theory and Applications.* In: World Scientific, 2014. ISBN 978-9814590099.

[24] FAWCETT, Tom. An introduction to ROC analysis. In: *Pattern Recognition Letters* [online]. 2006, 27(8), s. 861-874 [cit.2016-04-18]. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S016786550500303X

[25] BRADLEY, Andrew P. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms.In: *Pattern Recognition.* 30(6), s. 1145–1159.

[26] PLATT, John C. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* Microsoft Research Technical Report MSR-TR-98-14, 1998. Dostupné z: http://research.microsoft.com/ jplatt/smoTR.pdf

Seznam obrázků

Seznam tabulek

Seznam algoritmů

Seznam příloh

Příloha A : Výsledky klasifikace pro softmax normalizovaná data

Veškeré hodnoty v tabulkách výsledků klasifikace při testování pomocí hold-out validace a 10-násobné CV jsou ve tvaru „průměr ± směrodatná odchylka“. Při testování pomocí LOOCV směrodatná odchylka byla nulová.

Tabulka 7: Zprůměrované hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí hold-out validace

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	9,8	4,2	11,0	3,0	10,0	4,0	10,0	4,0
	1,4	12,6	1,4	13,6	1,8	12,2	1,7	12,3
NN	11,5	2,5	10,6	3,4	11,4	2,6	11,3	2,7
	1,9	12,1	10,4	3,6	2,3	11,7	2,0	12,0
SVM	11,4	2,6	9,7	4,3	11,4	2,6	11,6	2,4
	2,0	12,0	3,5	10,5	1,9	12,1	2,1	11,9
NBK	8,9	5,1	5,0	9,0	8,9	5,1	9,0	5,0
	1,3	12,7	1,2	12,9	1,1	12,9	1,4	12,6
DT	11,6	2,4	10,0	4,0	11,8	2,2	11,8	2,2
	1,1	12,9	3,6	11,4	1,2	12,8	1,4	12,6

Tabulka 8: Výsledky klasifikace pro jednotlivé algoritmy,testované pomocí hold-out validace

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS *k*-NN	70,1 ± 12,0	89,71 ± 7,95	87,7 ± 9,0	75,7 ± 8,1	79,9 ± 7,4
	SFS SVM	78,6 ± 11,2	74,1 ± 12,1	76,1 ± 8,9	78,5 ± 8,9	76,3 ± 7,1
	ReliefF	71,1 ± 12,8	87,4 ± 9,3	85,6 ± 8,9	76,0 ± 8,6	79,3 ± 7,6
	Celá tabulka	71,7 ± 11,6	88,1 ± 8,6	86,5 ± 8,8	76,4 ± 8,0	79,9 ± 6,8
NN	SFS *k*-NN	82,1 ± 10,1	86,2 ± 9,4	86,3 ± 8,4	83,5 ± 8,1	84,2 ± 6,7
	SFS SVM	75,7 ± 11,3	74,2 ± 13,0	75,6 ± 9,4	76,0 ± 9,0	75,0 ± 7,8
	ReliefF	81,5 ± 9,6	83,6 ± 11,6	84,4 ± 9,0	82,5 ± 7,5	82,5 ± 6,6
	Celá tabulka	80,9 ± 11,1	85,4 ± 9,5	85,4 ± 8,4	82,6 ± 8,8	83,1 ± 6,8
SVM	SFS *k*-NN	81,7 ± 10,7	85,6 ± 8,5	85,5 ± 7,3	83,2 ± 8,4	83,6 ± 6,4
	SFS SVM	69,0 ± 13,3	74,8 ± 14,3	74,5 ± 10,4	71,5 ± 8,9	71,9 ± 8,4
	ReliefF	81,2 ± 10,8	86,3 ± 9,1	86,2 ± 8,3	82,9 ± 8,2	83,8 ± 6,8
	Celá tabulka	82,6 ± 9,3	84,9 ± 8,72	85,0 ± 7,9	83,5 ± 7,9	83,8 ± 6,7
NBK	SFS *k*-NN	63,6 ± 12,8	91,1 ± 8,1	88,6 ± 9,3	72,2 ± 7,4	77,3 ± 6,8
	SFS SVM	35,4 ± 15,9	91,8 ± 8,8	84,5 ± 15,3	59,3 ± 6,3	63,6 ± 7,1
	ReliefF	65,4 ± 13,8	90,6 ± 8,8	88,7 ± 9,2	73,2 ± 8,0	78,0 ± 6,9
	Celá tabulka	63,7 ± 12,7	90,4 ± 8,7	88,4 ± 9,4	72,1 ± 6,7	77,0 ± 6,0
DT	SFS *k*-NN	82,9 ± 11,8	91,9 ± 7,1	91,5 ± 7,0	85,1 ± 8,9	87,4 ± 6,9
	SFS SVM	72,8 ± 7,6	71,4 ± 12,2	74,7 ± 10,4	73,0 ± 8,6	72,8 ± 7,6
	ReliefF	83,9 ± 10,8	91,1 ± 8,3	91,3 ± 7,0	86,0 ± 7,8	87,5 ± 5,2
	Celá tabulka	84,3 ± 11,8	90,1 ± 9,1	90,3 ± 7,6	86,2 ± 8,3	87,2 ± 6,1

Tabulka 9: Průměrné hodnoty AUC při testování hold-out validací

Algoritmus	SFS *k*-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,925	0,924	0,927	0,934
NN	0,991	0,991	0,991	0,991
SVM	0,962	0,897	0,959	0,962
NBK	0,970	0,899	0,973	0,969
DT	0,991	0,967	0,995	0,993

Tabulka 10: Zprůměrované hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí 10-fold CV

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	32,2	10,8	35,1	7,9	31,0	12,0	32,4	10,6
	5,0	38,0	11,0	32,0	6,8	36,2	5,3	37,7
NN	36,8	6,2	32,3	10,7	35,3	7,7	36,3	6,7
	5,6	37,4	10,1	32,9	6,5	36,5	6,2	36,8
SVM	35,7	7,3	30,5	12,5	35,1	7,9	37,1	5,9
	6,8	36,2	10,8	32,2	5,2	37,8	6,1	36,9
NBK	26,6	16,4	12,6	30,4	26,1	16,9	26,0	17,0
	2,9	40,1	2,5	40,5	2,4	40,6	2,9	40,1
DT	37,3	5,7	30,8	12,2	38,4	4,6	37,7	5,3
	4,2	38,8	11,9	31,1	4,2	38,8	4,4	38,6

Tabulka 11: Výsledky klasifikace pro jednotlivé algoritmy, testované pomocí 10-fold CV

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS k-NN	74,9 ± 1,8	88,3 ± 2,4	86,6 ± 2,4	77,9 ± 1,3	81,6 ± 1,5
	SFS SVM	81,6 ± 2,9	74,5 ± 2,4	76,2 ± 1,8	80,3 ± 2,5	78,1 ± 1,8
	ReliefF	72,0 ± 2,5	84,1 ± 2,9	82,0 ± 2,7	75,1 ± 1,8	78,1 ± 1,9
	Celá tabulka	75,3 ± 2,6	87,7 ± 2,6	86,1 ± 2,5	78,1 ± 1,8	81,2 ± 1,8
NN	SFS k-NN	85,6 ± 2,4	87,0 ± 2,7	86,8 ± 2,5	85,8 ± 2,1	92,8 ± 2,2
	SFS SVM	75,1 ± 5,0	76,5 ± 3,9	76,2 ± 3,8	75,5 ± 4,2	81,5 ± 4,1
	ReliefF	82,1 ± 1,6	84,9 ± 3,3	84,6 ± 2,8	82,6 ± 1,1	89,8 ± 1,5
	Celá tabulka	84,4 ± 2,9	85,6 ± 3,6	85,5 ± 3,2	84,6 ± 2,7	91,4 ± 2,8
SVM	SFS k-NN	82,0 ± 2,6	84,3 ± 2,2	84,1 ± 1,9	83,2 ± 2,2	82,5 ± 0,0
	SFS SVM	70,9 ± 3,2	75,0 ± 3,1	73,9 ± 2,7	72,1 ± 2,5	72,9 ± 2,4
	ReliefF	81,6 ± 2,5	87,9 ± 2,7	87,1 ± 2,5	82,7 ± 2,0	84,7 ± 1,8
	Celá tabulka	86,3 ± 2,4	85,8 ± 2,0	85,9 ± 1,7	86,3 ± 2,1	86,0 ± 1,5
NBK	SFS k-NN	61,8 ± 2,2	93,3 ± 1,9	90,3 ± 2,5	70,1 ± 1,3	77,5 ± 1,4
	SFS SVM	29,2 ± 2,55	94,3 ± 1,6	83,7 ± 4,1	57,1 ± 1,0	61,8 ± 1,6
	ReliefF	60,7 ± 2,5	94,3 ± 1,4	91,5 ± 1,9	70,6 ± 1,4	77,5 ± 1,4
	Celá tabulka	60,4 ± 2,2	93,3 ± 1,9	90,0 ± 2,5	70,2 ± 1,2	76,8 ± 1,4
DT	SFS k-NN	86,8 ± 3,6	90,3 ± 3,0	90,1 ± 2,7	87,4 ± 2,9	88,5 ± 2,0
	SFS SVM	71,7 ± 3,5	72,3 ± 4,3	72,2 ± 3,1	71,9 ± 2,6	72,0 ± 2,6
	ReliefF	89,4 ± 3,0	90,3 ± 2,2	90,2 ± 2,1	89,6 ± 2,7	89,8 ± 1,2
	Celá tabulka	87,7 ± 3,5	89,7 ± 2,8	89,6 ± 2,5	88,1 ± 3,1	88,7 ± 2,2

Tabulka 12: Průměrné hodnoty AUC při testování pomocí 10-fold CV

Algoritmus	SFS k-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,921	0,949	0,937	0,955
NN	0,994	0,994	0,994	0,994
SVM	0,959	0,886	0,956	0,955
NBK	0,960	0,893	0,965	0,959
DT	0,987	0,984	0,998	0,998

Tabulka 13: Hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí LOOCV

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	32	11	35	8	33	10	33	10
	6	37	13	30	7	36	6	37
NN	37	6	33	10	36	7	35	8
	6	37	11	32	6	37	9	34
SVM	35	8	31	12	34	9	37	6
	6	37	10	33	5	38	5	38
NBK	27	16	14	29	28	15	25	18
	3	40	2	41	2	41	3	40
DT	36	7	29	14	39	4	36	7
	4	39	12	31	4	39	4	39

Tabulka 14: Výsledky klasifikace pro jednotlivé algoritmy, testované pomocí LOOCV

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS *k*-NN	74,4	86,0	84,2	77,1	80,2
	SFS SVM	81,4	69,8	72,9	78,9	75,6
	ReliefF	76,7	83,7	82,5	78,3	80,2
	Celá tabulka	76,7	86,0	84,6	78,7	81,4
NN	SFS *k*-NN	86,0	86,0	86,0	86,0	86,0
	SFS SVM	76,7	76,4	75,0	76,2	75,6
	ReliefF	83,7	86,0	85,7	84,1	84,9
	Celá tabulka	81,4	79,1	79,5	81,0	80,2
SVM	SFS *k*-NN	81,4	86,0	85,4	82,2	83,7
	SFS SVM	72,1	76,7	75,6	73,3	74,4
	ReliefF	79,1	88,4	87,2	80,8	83,7
	Celá tabulka	86,0	88,4	88,1	86,4	87,2
NBK	SFS *k*-NN	62,8	93,0	90,0	71,4	77,9
	SFS SVM	32,6	95,3	87,5	58,6	64,0
	ReliefF	65,1	95,3	93,3	73,2	80,2
	Celá tabulka	58,1	93,0	89,3	69,0	75,6
DT	SFS *k*-NN	83,7	90,7	90,0	84,8	87,2
	SFS SVM	67,4	72,1	70,7	68,9	69,8
	ReliefF	90,7	90,7	90,7	90,7	90,7
	Celá tabulka	83,7	90,7	90,0	84,8	87,2

Tabulka 15: Hodnoty AUC při testovaní pomocí LOOCV

Algoritmus	SFS *k*-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,938	0,926	0,937	0,955
NN	0,994	0,995	0,994	0,994
SVM	0,959	0,886	0,956	0,955
NBK	0,960	0,863	0,965	0,959
DT	0,987	0,984	0,998	0,998

Příloha B : Výsledky klasifikace pro min-max normalizovaná data

Tabulka 16: Zprůměrované hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí hold-out validace

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	9,8	4,2	9,3	4,7	10,6	3,4	10,3	3,7
	2,8	11,2	2,6	11,4	3,2	10,8	1,7	12,3
NN	10,2	3,8	9,6	4,4	11,7	2,3	11,3	2,7
	4,0	10,0	4,0	10,0	2,6	11,4	2,5	11,5
SVM	9,5	4,5	9,3	4,7	11,1	2,9	11,3	2,7
	3,0	11,0	2,6	11,4	2,0	12,0	2,2	11,8
NBK	5,3	8,7	7,2	6,8	7,1	6,9	8,1	5,9
	1,7	12,3	1,2	12,8	0,8	13,2	1,1	12,9
DT	10,4	3,6	10,9	3,1	11,8	2,2	11,8	2,2
	3,9	10,1	3,5	10,5	1,1	12,9	1,3	12,7

Tabulka 17: Výsledky klasifikace pro jednotlivé algoritmy, testované pomocí hold-out validace

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS k-NN	70,1 ± 11,0	80,1 ± 10,3	78,6 ± 9,3	73,3 ± 7,5	75,1 ± 7,3
	SFS SVM	66,6 ± 11,8	81,7 ± 9,2	79,0 ± 9,3	71,6 ± 7,9	74,1 ± 7,4
	ReliefF	75,8 ± 12,9	77,0 ± 11,1	77,3 ± 9,4	76,9 ± 9,8	76,4 ± 8,5
	Celá tabulka	73,7 ± 12,8	87,8 ± 8,3	86,2 ± 9,0	77,7 ± 9,1	80,8 ± 8,1
NN	SFS k-NN	72,6 ± 12,0	71,6 ± 13,5	73,0 ± 9,5	73,1 ± 9,7	72,1 ± 7,4
	SFS SVM	68,9 ± 12,9	71,5 ± 11,9	71,4 ± 9,4	70,5 ± 9,3	70,2 ± 8,0
	ReliefF	83,9 ± 9,1	81,6 ± 10,9	82,8 ± 8,8	84,1 ± 7,8	82,7 ± 6,7
	Celá tabulka	80,9 ± 8,5	82,5 ± 10,6	83,0 ± 8,7	81,6 ± 7,32	81,7 ± 6,9
SVM	SFS k-NN	67,9 ± 11,1	78,8 ± 9,4	76,9 ± 8,1	71,7 ± 7,1	73,4 ± 6,3
	SFS SVM	66,3 ± 10,7	81,4 ±9,0	78,7 ± 8,6	71,2 ± 6,7	73,9 ± 6,7
	ReliefF	79,2 ± 11,3	86,0 ± 9,4	85,8 ± 8,6	81,4 ± 8,5	82,6 ± 6,8
	Celá tabulka	80,7 ± 10,7	84,5 ± 8,7	84,5 ± 7,6	82,3 ± 8,3	82,6 ± 6,1
NBK	SFS k-NN	37,7 ± 17,0	87,8 ± 9,5	77,4 ± 15,3	59,2 ± 7,3	62,8 ± 8,0
	SFS SVM	51,1 ±15,0	91,4 ± 8,4	87,8 ± 10,8	66,0 ± 7,1	71,3 ± 6,3
	ReliefF	50,4 ± 18,5	94,4 ± 6,4	91,1 ± 9,4	66,7 ± 8,8	72,4 ± 8,6
	Celá tabulka	57,7 ± 17,7	92,0 ± 7,7	88,7 ±9,6	69,8 ± 9,4	74,9 ± 8,2
DT	SFS k-NN	74,3 ± 11,1	72,1 ± 13,7	73,8 ± 9,7	74,4± 8,6	73,2 ± 7,8
	SFS SVM	78,1 ± 13,5	75,1 ±11,7	76,6 ±8,8	78,8 ± 10,3	76,6 ± 7,7
	ReliefF	84,3 ± 10,6	91,9 ±7,1	91,8 ± 6,4	86,3 ± 8,0	88,1 ± 5,4
	Celá tabulka	84,3 ± 8,8	91,0 ± 7,2	90,9 ± 6,6	85,8 ± 7,0	87,6 ± 5,2

Tabulka 18: Průměrné hodnoty AUC při testování hold-out validací

Algoritmus	SFS k-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,908	0,895	0,934	0,920
NN	0,992	0,992	0,992	0,992
SVM	0,854	0,859	0,949	0,954
NBK	0,919	0,936	0,938	0,974
DT	0,977	0,977	0,990	0,995

Tabulka 19: Zprůměrované hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí 10-fold CV

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	31,9	11,1	29,2	13,8	33,7	9,3	32,4	10,6
	7,5	35,5	6,8	36,2	11,2	31,8	5,8	37,2
NN	31,4	11,6	29,4	13,6	35,1	7,9	33,3	9,7
	12,6	30,4	13,0	30,0	8,1	34,9	7,9	35,1
SVM	29,2	13,8	28,7	14,3	34,5	8,5	37,2	5,8
	10,4	32,6	7,3	35,7	5,4	37,6	6,2	36,8
NBK	12,4	30,6	20,7	22,3	21,5	21,5	23,0	20,0
	4,0	39,0	2,1	40,9	2,1	40,9	2,5	40,5
DT	31,4	11,6	34,2	8,8	38,2	4,8	37,7	5,3
	10,9	32,2	9,8	33,2	4,0	39,0	4,3	38,7

Tabulka 20: Výsledky klasifikace pro jednotlivé algoritmy, testované pomocí 10-fold CV

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS *k*-NN	74,2 ± 2,9	82,7 ± 2,0	81,0 ± 2,0	76,2 ± 2,2	78,4 ± 1,3
	SFS SVM	67,8 ± 3,6	84,3 ± 1,6	81,2 ± 1,8	72,4 ± 2,2	76,0 ± 1,9
	ReliefF	78,4 ± 3,2	73,9 ± 3,0	75,0 ± 2,2	77,4 ± 2,7	76,1 ± 2,1
	Celá tabulka	75,3 ± 1,7	86,6 ± 1,7	85,0 ± 1,8	77,9 ± 1,3	81,0 ± 1,3
NN	SFS *k*-NN	73,0 ± 4,8	70,7 ± 3,7	71,4 ± 3,1	72,5 ± 4,0	77,3 ± 3,5
	SFS SVM	68,4 ± 5,5	69,8 ± 4,2	69,4 ± 3,6	68,9 ± 4,1	74,3 ± 3,9
	ReliefF	81,6 ± 3,0	81,2 ± 3,7	81,4 ± 2,8	81,6 ± 1,9	87,5 ± 1,6
	Celá tabulka	77,4 ±4,8	81,6 ± 5,0	80,9 ± 4,5	78,4 ± 4,3	85,5± 4,4
SVM	SFS *k*-NN	67,9 ± 2,4	75,9 ± 2,4	73,9 ± 2,1	70,3 ± 1,8	71,9 ± 1,8
	SFS SVM	66,8 ± 2,7	83,0 ± 2,0	79,8 ± 2,1	71,5 ± 1,8	74,9 ± 1,8
	ReliefF	80,5 ± 2,8	87,5 ± 2,6	86,5 ± 2,5	81,6 ± 2,9	83,8 ± 2,4
	Celá tabulka	86,4 ± 2,9	85,6 ± 2,6	85,7 ± 2,0	86,3 ± 2,6	86,0 ± 2,0
NBK	SFS *k*-NN	28,8 ± 2,7	90,7 ± 2,3	75,9 ± 4,4	56,1 ± 0,9	59,8 ± 1,5
	SFS SVM	48,0 ± 1,9	95,2 ± 2,2	91,0 ± 3,7	64,7 ± 1,0	71,6 ± 1,5
	ReliefF	46,9 ± 3,5	95,0 ± 3,5	91,0 ± 2,0	65,5 ± 1,6	72,5 ± 1,9
	Celá tabulka	53,6 ± 2,8	94,2 ± 1,8	90,2 ± 2,7	67,0 ± 1,4	73,9 ± 1,7
DT	SFS *k*-NN	77,4 ± 4,8	73,3 ± 4,3	73,1 ± 3,4	72,8 ± 3,6	72,9 ± 3,2
	SFS SVM	77,9 ± 3,9	77,0 ± 4,5	77,3 ± 3,6	77,7 ± 3,2	77,4 ± 2,9
	ReliefF	89,2 ± 3,8	90,7 ± 1,9	90,5 ±1,8	89,5 ± 3,4	89,9 ± 2,1
	Celá tabulka	87,2 ± 3,1	90,0 ± 2,3	89,8 ± 2,1	87,6 ± 2,6	88,6 ± 1,8

Tabulka 21: Průměrné hodnoty AUC při testování 10-fold CV

Algoritmus	SFS *k*-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,897	0,872	0,929	0,932
NN	0,994	0,995	0,994	0,995
SVM	0,855	0,860	0,944	0,948
NBK	0,908	0,929	0,970	0,970
DT	0,994	0,991	0,987	0,998

Tabulka 22: Hodnoty matic záměn pro jednotlivé klasifikátory, testované pomocí LOOCV

Algoritmus	SFS k-NN		SFS SVM		ReliefF		Celá tabulka	
k-NN	30	13	29	14	36	7	33	10
	7	36	7	36	12	31	6	37
NN	27	16	31	12	36	7	34	9
	12	31	15	28	9	34	7	36
SVM	30	13	28	15	33	10	37	6
	12	31	7	36	6	37	6	37
NBK	11	32	21	22	22	21	22	21
	5	38	1	42	2	41	2	41
DT	28	15	35	8	39	4	36	7
	9	34	8	35	4	39	4	39

Tabulka 23: Výsledky klasifikace pro jednotlivé algoritmy, testované pomocí LOOCV

Algoritmus	Data	Senzitivita (%)	Specificita (%)	PPV (%)	NPV (%)	Úspěšnost (%)
k-NN	SFS k-NN	69,8	83,7	81,1	73,5	76,7
	SFS SVM	67,4	83,7	80,6	72,0	75,6
	ReliefF	83,7	72,1	75,0	81,6	77,9
	Celá tabulka	76,7	86,0	84,6	78,7	81,4
NN	SFS k-NN	62,8	72,1	69,2	66,0	67,4
	SFS SVM	72,1	65,1	67,4	70,0	68,6
	ReliefF	83,7	79,1	80,0	83,0	81,4
	Celá tabulka	79,1	83,7	83,0	80,0	81,4
SVM	SFS k-NN	69,8	72,1	71,4	70,4	70,9
	SFS SVM	65,1	83,7	80,0	70,6	74,4
	ReliefF	76,8	86,0	84,6	78,7	81,4
	Celá tabulka	86,0	86,0	86,0	86,0	86,0
NBK	SFS k-NN	25,6	88,4	68,8	54,3	57,0
	SFS SVM	48,8	97,7	95,5	62,6	73,3
	ReliefF	51,2	95,3	91,7	66,1	73,3
	Celá tabulka	51,2	95,3	91,7	66,1	73,3
DT	SFS k-NN	65,1	79,1	75,7	69,4	72,1
	SFS SVM	81,4	81,4	81,4	81,4	81,4
	ReliefF	90,1	90,1	90,1	90,1	90,1
	Celá tabulka	83,7	90,7	90,0	84,8	87,2

Tabulka 24: Hodnoty AUC při testovaní pomocí LOOCV

Algoritmus	SFS k-NN	SFS SVM	ReliefF	Celá tabulka
k-NN	0,897	0,872	0,929	0,932
NN	0,995	0,995	0,995	0,995
SVM	0,855	0860,	0,944	0,948
NBK	0,908	0,929	0,970	0,970
DT	0,994	0,991	0,987	0,998

Příloha C : Obsah CD

- klíčová slova.pdf
- abstrakt (česky).pdf
- abstrakt (anglicky).pdf
- zadání.pdf
- bakalářská práce.pdf
- zdrojové kódy
- data