STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor: 1. Matematika a statistika

Identifikace typu tkáně pomocí Ramanovy spektroskopie

Nikolas Pippal

Olomouc 2023

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

IDENTIFIKACE TYPU TKÁNĚ POMOCÍ RAMANOVY SPEKTROSKOPIE

TISSUE TYPE IDENTIFICATION USING RAMAN SPECTROSCOPY

AUTOR	Nikolas Pippal
ŠKOLA	Gymnázium, Olomouc - Hejčín
KRAJ	Olomoucký
ŠKOLITEL	RNDr. Ondřej Pavlačka, Ph.D.
OBOR	1. Matematika a statistika

Olomouc 2023

Prohlášení

Prohlašuji, že svou práci na téma *Identifikace typu tkáně pomocí Ramanovy spektroskopie* jsem vypracoval/a samostatně pod vedením RNDr. Ondřeje Pavlačky, Ph.D. a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Dále prohlašuji, že tištěná i elektronická verze práce SOČ jsou shodné a nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a změně některých zákonů (autorský zákon) v platném znění.

V Olomouci dne: _____ _____

Nikolas Pippal

Poděkování

Mé díky patří panu RNDr. Ondřeji Pavlačkovi, Ph.D., za vedení práce a odborné rady, a také Přírodovědecké fakultě Univerzity Palackého v Olomouci za příležitost se na tomto projektu podílet. Také bych chtěl poděkovat doc. RNDr. Václavu Rancovi, Ph.D., za konzultace a veškerá data, bez kterých by tato práce nebyla. Dále bych chtěl poděkovat Mgr. Šárce Richterkové, bez které bych o této možnosti nevěděl. Velký dík patří také mé rodině za velkou podporu a trpělivost, kterou se mnou měla.

Anotace

Tato práce se zaměřuje na normalizaci, transformaci a klasifikaci dat z Ramanovy spektroskopie mozkové tkáně s cílem rozlišit mezi mozkem a nádorem. Stěžejním krokem při předzpracování dat byla baseline korekce. Pro klasifikaci byly využity klasifikátory k-nearest neighbors (KNN) a rozhodovací strom.

Klíčová slova

Ramanova spektroskopie; transformace dat; klasifikace; baseline korekce; nádor

Annotation

This thesis focuses on the normalization, transformation and classification of Raman spectroscopy data of brain tissue in order to distinguish between brain and tumor. A key step in the data preprocessing was baseline correction. The k-nearest neighbors (KNN) classifiers and decision tree were used for classification.

Keywords

Raman spectroscopy; data transformation; classification; baseline correction; tumour

Obsah

Úvod

Gliomy představují více než třetinu všech intrakraniálních nádorů. Jedná se o heterogenní skupinu primárních mozkových nádorů s různorodým biologickým chováním, která významně ovlivňuje přežití a kvalitu života pacientů. I přes nedávný pokrok v diagnostických a terapeutických možnostech je prognóza pacientů s gliovými nádory stále nepříznivá. Základem terapie je radikální bezpečná resekce.[1] K dosažení maximální radikální resekce se používá řada technik[2]:

- Standardní resekce pomocí mikroskopických technik
- Neuronavigace s využitím předoperačního anatomického plánování
- Funkční navigace (fMRI, traktografie)
- Fluorescenčníperioperačně naváděná resekce (ALA, fluorescein)

Zcela novým přístupem umožňujícím průběžnou perioperační kontrolu hranice resekce je využití Ramanovy spektroskopie, která na základě spektrální analýzy hodnotí biochemické složení tkáně. Spektrum je pro každou tkáň jedinečné a umožňuje rozlišit nádorovou a zdravou tkáň v minimálním vzorku odpovídajícímu hrotu mikrosondy.

Cílem této práce je spolehlivá identifikace nádorové tkáně v mozku živých pacientů pomocí matematického/statistického zpracování dat. Za tímto účelem budou získána Ramanova spektra mozkové tkáně se známými histopatologickými znaky. V první části se zaměříme na původ dat pro klasifikaci. Obsahem druhé části práce jsou možné transformační metody pro přesnější klasifikaci a na závěr vyzkoušíme různé klasifikátory a zhodnotíme výsledky.

Kapitola 1

Původ dat

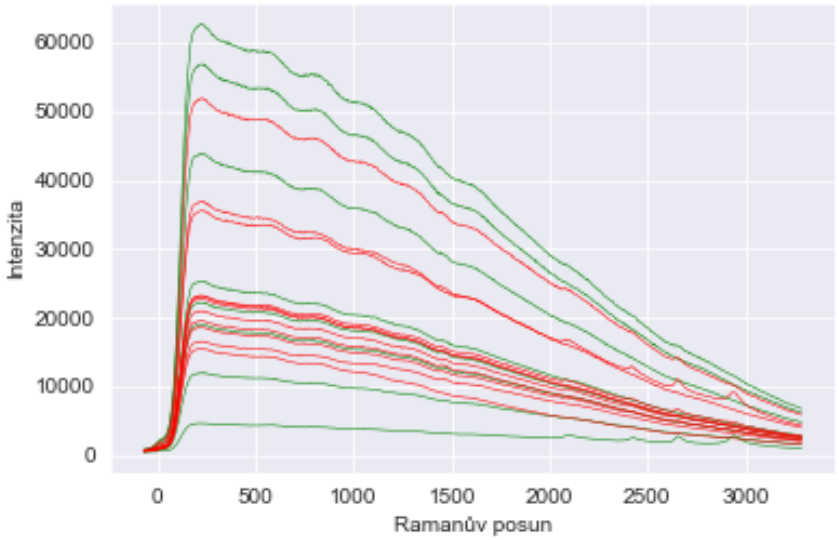
Pro tento projekt jsou data získávaná při operaci pacientů s mozkovým nádorem pomocí optické vláknové Ramanovy sondy, která funguje na principu Ramanovy spektroskopie. Ramanova spektroskopie je metoda, která umožňuje studovat vnitřní strukturu molekul v materiálech, jako jsou buňky nebo tkáně. K tomu se používá laserový paprsek, který interaguje s molekulami a měří se spektrum záření, které je v důsledku této interakce vydáváno. Toto spektrum obsahuje informace o vibračních pohybech molekul v materiálu a umožňuje identifikovat různé druhy molekul a sledovat změny v jejich struktuře.

Ramanova spektroskopie[3] je často používána v biologických výzkumech k analýze tkání nebo krevních vzorků. Je to kvůli tomu, že může identifikovat specifické vibrační módy, které jsou citlivé na různé fyzikálně-chemické faktory, jako je teplota nebo pH. Tyto vibrační módy se nazývají Ramanovy markery a jsou důležité pro identifikaci různých druhů biomolekul a sledování biochemických procesů, jako jsou fáze buněčného cyklu nebo stádia rakoviny.

Ve své práci jsem pracoval s celkem 19 pacienty, od kterých jsem měl k dispozici vzorky. Každý vzorek obsahoval zhruba 12 měření, přičemž většina z nich se skládala ze 2 až 4 spekter mozku a zbylá spektra představovala nádorové tkáně. Celkem jsem tedy pracoval s 222 spektry, z nichž bylo 50 mozkových a zbylých 172 nádorových. Tři měření jsem musel ručně odstranit, jelikož se chovala velmi odlišně. Každé spektrum sestává z uspořádané n-tice čísel, která odpovídá hodnotám Ramanova posunu, a dále n-tice čísel odpovídající hodnotám Ramanovy spektroskopie pro každou hodnotu Ramanova posunu.

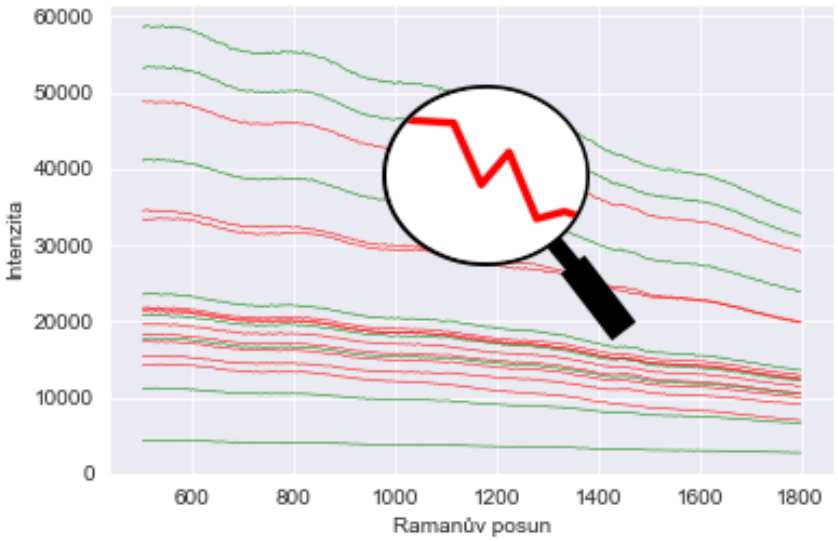
Před analýzou každého spektra jsem získal informaci od patologa, zda se jedná o spektrum mozku nebo nádoru.

Před úpravou dat je vhodné data nejdříve vizualizovat, abychom věděli, jak vypadají. V Grafu 1.1 můžeme vidět tato surová data, kde zeleně jsou vykresleny výsledky měření zdravého mozku a červeně měření nádoru.



Graf 1.1: Surová data

Tato data je možné analyzovat pouze na intervalu $500\,cm^{-1}$ až $1800\,cm^{-1}$. Dolní i horní ohraničení datové sady bylo určeno odborníkem, který data poskytl. Data již nelze analyzovat za hranicí $1800\,cm^{-1}$, protože měření je zde ovlivněno okolním světlem.



Graf 1.2: Ořezaná data

Jak si můžeme všimnout v Grafu 1.2, měření je velice neuhlazené. Tuto neuhlazenost je potřeba odstranit pomocí jedné z metod popsaných v následující kapitole.

Kapitola 2

Transformační metody

Samotné surové datové soubory z Ramanovy spektroskopie mohou být velmi složité a obsáhlé. Před samotnou klasifikací je tedy nutné data transformovat, abychom získali relevantní informace, které lze použít pro klasifikaci. Transformace dat se obvykle provádí s cílem odstranit šum a normalizovat data. Všechny následující transformace byly provedeny pomocí programovacího jazyku *Python* s použitím knihoven $NumPy$, $Pandas$, $SciPy$, $Scikit-learn$ a dále knihoven $Matplotlib$ a *Plotly* pro vykreslování grafů.

2.1 Metody vyhlazení dat

Metody vyhlazení dat jsou užitečné nástroje při analýze biologických dat. Tyto metody slouží k redukci šumu a odstranění náhodných fluktuací, což umožňuje lepší detekci skutečných biologických signálů. Při práci s biologickými daty jsou fluktuace běžné a mohou být způsobeny různými faktory. Použití vhodného vyhlazovacího algoritmu může vést k výraznému zlepšení kvality dat a umožnit přesnější analýzu a interpretaci biologických procesů.

2.1.1 Savitzky–Golay filtr

Savitzky-Golay filtr je numerická metoda pro hladké vyhlazení datových bodů.[4] Filtr funguje tak, že pro každý bod vstupních dat se vybere určitý počet sousedních bodů, tzv. okno, které jsou poté interpolovány polynomem nízkého stupně. Tento polynom je následně použit k aproximaci hodnoty v daném bodě. V Savitzky-Golay filtru se pro interpolaci bodů uvnitř okna využívá metoda nejmenších čtverců.

Výsledná úprava dat po použití této metody tedy závisí na dvou proměnných. Na parametru n, který odpovídá stupni aproximačního polynomu, a parametru w, kterým nastavíme šířku okna. Musí platit $w = 2H+1$, $H \in \mathbb{N}$, protože pro bod x_i musí existovat množina bodů $M = \{x_{i-H}, \ldots, x_i, x_{i+1}, \ldots, x_{i+H}\}$. Tím vznikají problémy na okrajích datové sady, kde neexistují všechny potřebné body. Tyto problémy můžeme vyřešit následným ořezáním datové sady. Můžeme si všimnout, že pokud jako stupeň polynomu nastavíme $n = 0$, již se nebude jednat o Savitzky-Golay filtr, ale o klouzavý průměr. Je třeba vhodně volit parametry filtru. Uvažujme následující situaci. Mějme okno velikosti $w = m$ a polynom stupně $n = m - 1$. Poté filtr nemá význam, jelikož polynom projde všemi body m. Pokud ale zároveň zvolíme příliš velkou velikost okna w a nízký stupeň polynomu n, může dojít ke zkresleni dat. Jelikož v tomto filtru pro interpolaci bodů uvnitř okna používáme metodu nejmenších čtverců, je předmětné si tuto metodu vysvětlit.

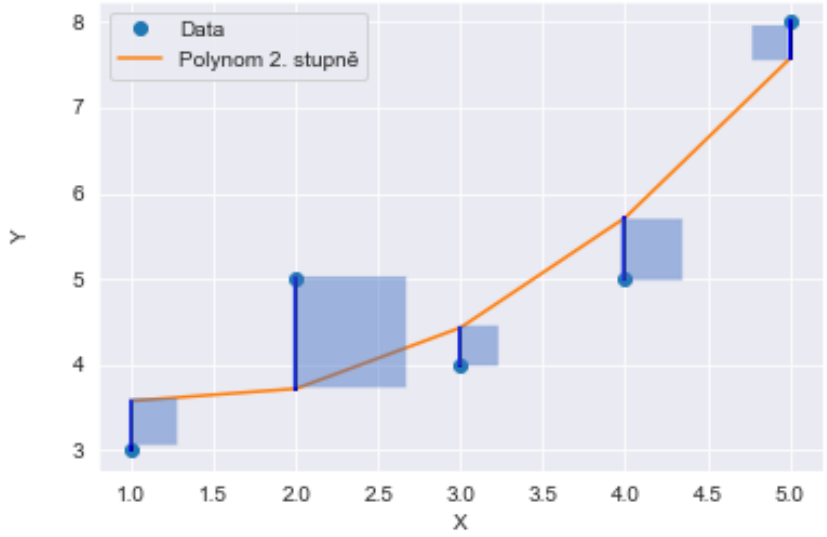
Metoda nejmenších čtverců

Mějme soubor dat $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, kde x_i jsou vstupní proměnné a y_i jsou odpovídající výstupy. Chceme nalézt funkci $f(x)$, která nejlépe aproximuje data. Funkce $f(x)$ může být libovolná spojitá funkce s definičním oborem odpovídajícímu datům.

Předpokládáme, že funkce $f(x)$ má tvar

$$f(x) = a_1 g_1(x) + a_2 g_2(x) + \ldots + a_m g_m(x),$$

kde $a_j \in \mathbb{R}$ jsou koeficienty, které určují váhu každé funkce $g_j(x)$ v lineární kombinaci. Funkce $g_j(x)$ mohou být libovolné funkce s definičním oborem odpovídajícímu datům. V praxi se obvykle volí jednoduché funkce, jako například polynomy, trigonometrické funkce nebo exponenciální funkce.



Graf 2.1: Aproximace bodů polynomem

Cílem je najít koeficienty a_j tak, aby funkce $f(x)$ co nejlépe aproximovala data. Pro metodu nejmenších čtverců se jako chyba aproximace volí suma čtverců odchylek mezi skutečnými hodnotami y_i a hodnotami aproximované funkce $f(x_i)$. Chceme tedy minimalizovat chybu aproximace, tzn.

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2.$$

Pro další výpočty můžeme minimalizovanou funkci přepsat do tvaru[5]

$$S = \frac{1}{2} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{m} a_j g_j(x_i))^2.$$

Toto je běžná forma nejmenších čtverců, cílem je přitom najít hodnoty $a_1, \ldots, a_m$, které minimalizují součet čtverců odchylek mezi předpovězenými hodnotami $\sum_{j=1}^{m} a_j g_j(x_i)$ a pozorovanými hodnotami y_i. Jedním ze způsobů nalezení minima je gradientní metoda, která spočívá v nalezení koeficientů a_j jako řešení systému rovnic

$$\frac{\partial S}{\partial a_j}(\bar{x}) = 0, j \in \{1, 2, \ldots, m\}.$$

Pro každé $j \in \{1, \ldots, m\}$ můžeme parciální derivaci $\frac{\partial S}{\partial a_j}(\bar{x})$ rozepsat, jako

$$\frac{\partial S}{\partial a_j}(\bar{x}) = -2 \sum_{i=1}^{n} (y_i - f(x_i)) g_j(x_i).$$

Dosazením této rovnosti do předchozího vztahu dostáváme

$$\begin{aligned} -2 \sum_{i=1}^{n} (y_i - f(x_i)) g_1(x_i) &= 0, \\ -2 \sum_{i=1}^{n} (y_i - f(x_i)) g_2(x_i) &= 0, \\ &\vdots \\ -2 \sum_{i=1}^{n} (y_i - f(x_i)) g_m(x_i) &= 0, \end{aligned}$$

což lze zkráceně zapsat ve tvaru

$$\sum_{i=1}^{n} (y_i - f(x_i)) g_j(x_i) = 0, \ j = 1, 2, 3, \ldots, m.$$

Tento výraz je vlastně systém m lineárních rovnic o m neznámých, který můžeme řešit například Gaussovou eliminací nebo QR rozkladem. Koeficienty a_j získáme jako řešení tohoto systému. Po nalezení koeficientů a_j můžeme aproximovat data pomocí funkce $f(x)$, kde

$$f(x) = a_1 g_1(x) + a_2 g_2(x) + \cdots + a_m g_m(x).$$

2.1.2 Klouzavé průměry

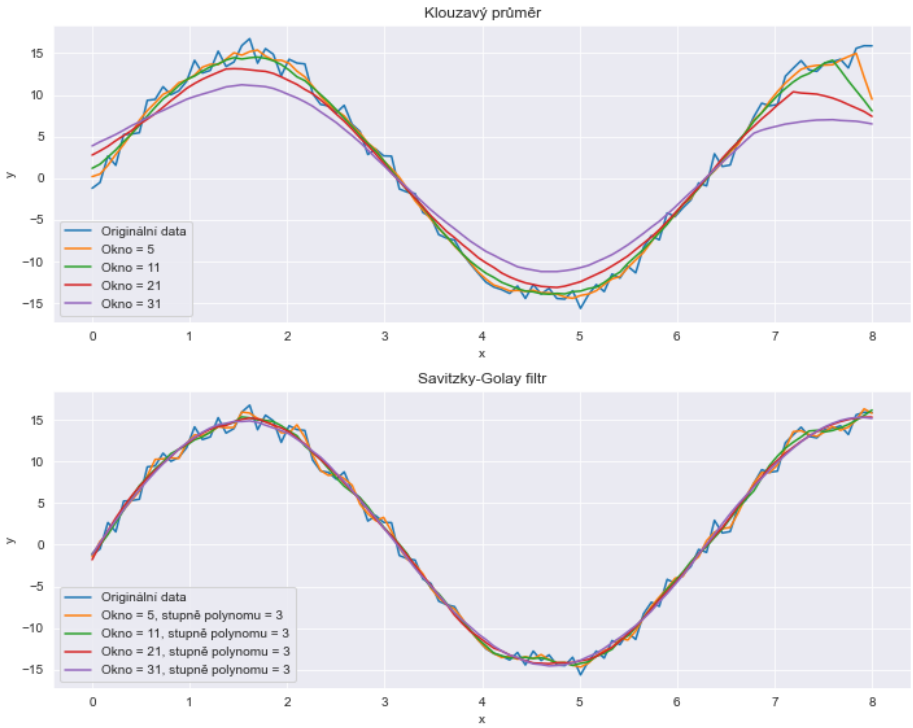
Jak už jsme si řekli, pokud jako stupeň polynomu v Savitzky-Golay filtru nastavíme $n = 0$, bude se jednat o klouzavý průměr. Klouzavý průměr je často používaná statistická metoda pro hladké vyhlazení datových řad. Tato metoda spočívá v průměrování hodnot v určitém okně kolem každého bodu v datové řadě. Šířka tohoto okna se obvykle nazývá délka klouzavého průměru.

Formálně lze definovat klouzavý průměr délky k pro datovou řadu $R = \{r_1, \ldots, r_n\}$ tak, že

$$x_i = \frac{1}{k} \sum_{j=i-(k-1)/2}^{i+(k-1)/2} x_j,$$

kde $i \in 1, 2, \ldots, n$ a k je liché číslo. Tedy klouzavý průměr $\hat{x}_i$ pro i-tý bod je průměrem k hodnot v okně s centrem v i-tém bodu.

Klouzavý průměr se často používá k vyhlazování biologických datových řad, kde hodnoty mohou být zatíženy náhodnými fluktuacemi a šumem. Použitím klouzavého průměru lze snížit tyto fluktuace a odhalit skryté trendy nebo obecné charakteristiky dat. Je však třeba být opatrný při volbě délky klouzavého průměru, protože příliš malá délka může vést k nedostatečnému vyhlazení a příliš velká délka může vést ke ztrátě specifik dat.



Graf 2.2: Porovnání parametrů

Jak lze vidět z Grafu 2.2, je velmi důležité správně zvolit parametry. Ve všech dalších úpravách dat budeme pracovat s daty, která jsme vyhladili pomoci Savitzky-Golay filtru s parametry $w = 15$ a $n = 3$. Tato kombinace se ukázala být nejlepší, jelikož data dostatečně vyhladila, ale zároveň zachovala skutečné chování dat.

2.2 Normování dat

Normování dat je proces úpravy hodnot v data setu s cílem převést je na stejný měřítkový rozsah. Cílem normování dat je zajistit, aby data byla přesnější pro analýzu a modelování. Používá se k tomu, aby bylo možné porovnávat nebo kombinovat data, která mají různé rozsahy. Normování dat také může pomoci snížit vliv extrémních hodnot v datovém setu a zlepšit výkon některých algoritmů strojového učení. Mezi běžně používané metody normování dat patří min-max scaling, z-score a unit length.

2.2.1 Min-max scaling

Když se na neupravená data podíváme, zjistíme, že všechna měření mají své maximum ve stejné hodnotě Ramanova posunu. To stejné platí i pro minima měření. Proto je vhodné data znormovat pomocí $Min - max\ scaling$ a tím dostat všechna měření na stejné měřítko. $Min - max\ scaling$[6] (také známý jako min-max normalizace) je metoda normování dat, která se používá k převodu hodnot v data setu na měřítko od 0 do 1.

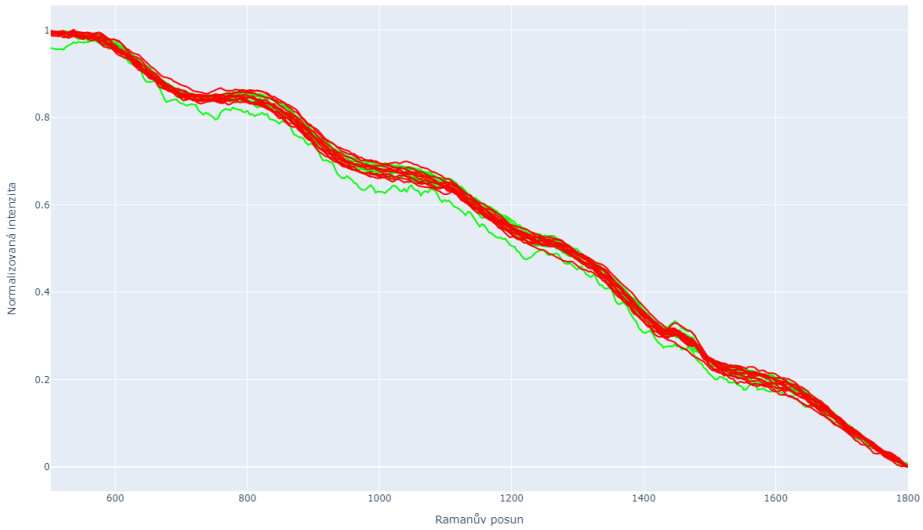
Metoda spočívá v přepočtu hodnot na nové hodnoty na základě minimální a maximální hodnoty v data setu. Pokud máme data set X s n hodnotami $x_1, \ldots, x_n$, tak nová normovaná hodnota x'_i se vypočítá

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)},$$

kde $\min(X)$ a $\max(X)$ jsou minimální a maximální hodnoty v data setu X. Tento výpočet umožňuje převést každou hodnotu v data setu na novou hod-

notu, která je v rozmezí od 0 do 1, kde $min(\tilde{X}) = 0$ a $max(\tilde{X}) = 1$

Výhodou min-max scalingu je, že zachovává relativní pozici jednotlivých hodnot v rámci data setu. Pokud jsou dvě hodnoty v data setu X ve vzdálenosti d, pak po $min - max\ scalingu$ budou tyto hodnoty vzdálené právě $\tilde{d}$, kde $\tilde{d} = \frac{d-min(X)}{max(X)-min(X)}$ To znamená, že $Min-max scaling$ zachovává uspořádání.



Graf 2.3: Min-max normalizace

V Grafu 2.3 můžeme vidět, že skutečně všechna měření jsou nyní ve stejném měřítku. Zároveň mají všechna spektra téměř identický průběh, čehož můžeme využít při následném transformování dat.

2.2.2 Unit length normalizace

Unit length normalizace je metoda, která se používá k převodu vektorů na vektory jednotkové délky. Tato metoda se často používá v oblastech, jako jsou strojové učení a analýza obrazu. Předpokládejme, že máme vektor $\mathbf{x} = [x_1, \ldots, x_n]$. Ten si můžeme představit jako bod v n-rozměrném prostoru.

Proč je však důležité normovat vektory? Jedním z důvodů je, že délka vektoru může ovlivnit jeho interpretaci. Například při použití klasifikátorů strojového učení může mít vektor s větší délkou větší vliv na klasifikaci než vektor s menší délkou. To může vést k nesprávným výsledkům klasifikace.

Unit length normalizace zajišťuje, že všechny vektory mají stejnou délku, což umožňuje přesnější porovnání vektorů. Tuto normalizaci můžeme provést dělením vektoru $\mathbf{x}$ jeho vlastní délkou, tzn.

$$\tilde{x} = \frac{\mathbf{x}}{||\mathbf{x}||},$$

kde $||\mathbf{x}||$ je euklidovská norma vektoru $\mathbf{x}$, definovaná jako

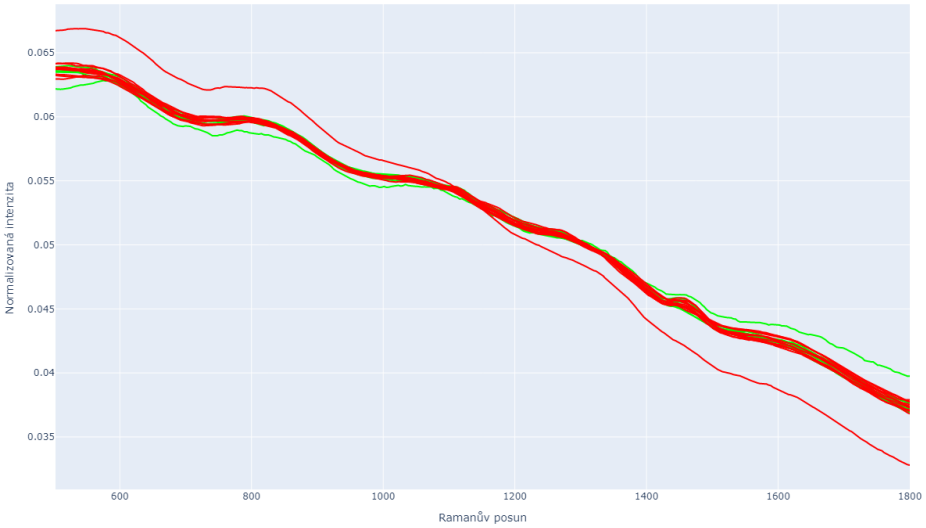
$$||\mathbf{x}|| = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Tato normalizace vede k vytvoření jednotkového vektoru, protože délka normovaného vektoru $\tilde{x}$ je rovna 1:

$$||\tilde{x}|| = \left|\left|\frac{\mathbf{x}}{||\mathbf{x}||}\right|\right| = \frac{||\mathbf{x}||}{||\mathbf{x}||} = 1$$

Jednotkové vektory mají výhodu v tom, že jsou invariantní vůči měřítku, což znamená, že jsou nezávislé na velikosti originálních dat a jsou vhodné pro použití v algoritmech, které se snaží nalézt podobnosti nebo rozdíly mezi vektory.

Po aplikaci unit length normalizace na naše data dostaneme Graf 2.4.



Graf 2.4: Jednotková vektorová délka

Opět můžeme vidět, že všechna spektra jsou nyní ve stejném měřítku a zároveň mají velmi podobný průběh. Pokud bychom na takto upravená data použili Min-max scaling, dostali bychom graf velice podobný Grafu 2.3.

2.2.3 Z-score

Z-score normalizace je statistická metoda, jež slouží k převodu dat na standardní normální rozdělení, které je definováno jako normální rozdělení se střední hodnotou 0 a směrodatnou odchylkou 1.[7] Směrodatná odchylka je statistická míra variability dat vzhledem k střední hodnotě a udává, jak moc jsou hodnoty v sadě dat rozprostřeny kolem střední hodnoty. Čím vyšší je směrodatná odchylka, tím více jsou hodnoty dat rozprostřeny a tím méně jsou koncentrované kolem střední hodnoty.

Směrodatnou odchylku můžeme vypočítat jako druhou odmocninu z variačního koeficientu, který se počítá jako průměr druhých mocnin odchylek

jednotlivých hodnot dat od střední hodnoty. Matematicky se to dá vyjádřit

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2},$$

kde σ značí směrodatnou odchylku, N značí počet prvků v sadě dat, x_i je i-tý prvek a μ je střední hodnota. Jako bodový odhad střední hodnoty dat μ můžeme použít výběrový průměr

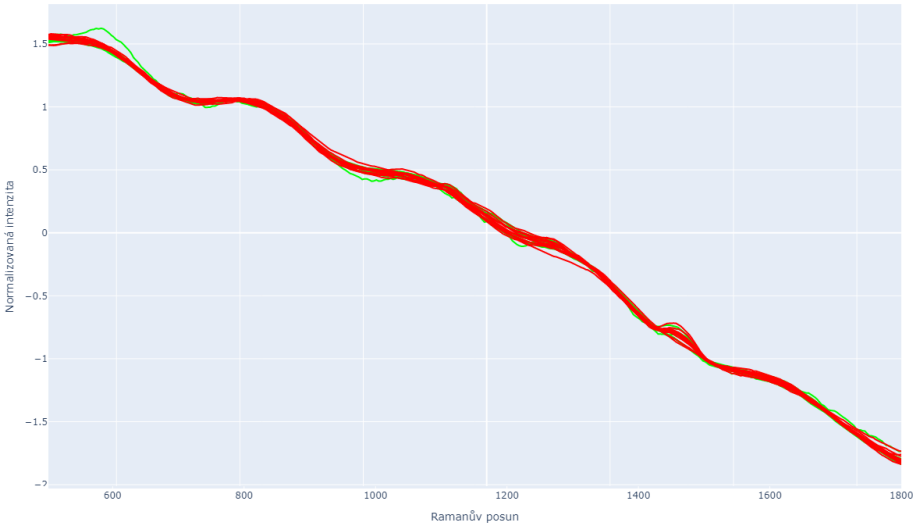
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Z-score normalizace umožňuje porovnávání a interpretaci různých souborů dat, protože výsledné hodnoty z-score normalizace jsou vyjádřeny v jednotkách směrodatné odchylky od střední hodnoty. Provedení Z-score normalizace zahrnuje výpočet z-score pro každý prvek v sadě dat. Z-score (také známý jako standardizované skóre) pro prvek x_i v sadě dat X s průměrem μ a směrodatnou odchylkou σ se vypočítá jako:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Tento výpočet převádí původní měření x_i na měření v jednotkách směrodatné odchylky od střední hodnoty. Pokud je z-score kladné, znamená to, že hodnota x_i je nad průměrem. Pokud je z-score záporné, znamená to, že hodnota x_i je pod průměrem.

Jak můžeme vidět v Grafu 2.5, všechna měření jsou převedena na stejné měřítko a také jejich průběh je téměř shodný.



Graf 2.5: Z-score normalizace

Nyní když máme data znormovaná pomocí tří různých metod, můžeme data začít transformovat. Při porovnání metod normalizace, které jsme využili, se ukazuje, že většinou vedou k podobným výsledkům. To znamená, že výsledky klasifikace dat nebudou výrazně ovlivněny tím, kterou metodu normalizace použijeme.

2.3 Transformování dat

Cílem této kapitoly je ukázat, že transformace dat je klíčová pro úspěšnou analýzu dat a pro dosažení přesných výsledků. V následujících sekcích se budeme podrobněji věnovat různým metodám transformace dat, které jsou často používány v analýze dat. Ukážeme, jak vybrat nejvhodnější transformace pro naše konkrétní data.

V této práci jsme již provedli normalizaci datové sady, ale pro úspěšnou analýzu mohou být nezbytné další transformace. Každá metoda klasifikace

má specifické požadavky na data, a proto je důležité provést transformace, které zajistí co nejlepší výkon.

V dalších sekcích této kapitoly se budeme podrobněji věnovat různým metodám transformace dat, které jsou často používány v analýze dat. Ukážeme, jak vybrat nejvhodnější transformace pro naše konkrétní datové sady.

2.3.1 Derivace

Když máme data z Ramanovy spektroskopie a chceme je zderivovat, můžeme použít diferenciální aproximaci. Pro získání derivace v bodě x_i můžeme použít následující aproximaci první derivace:

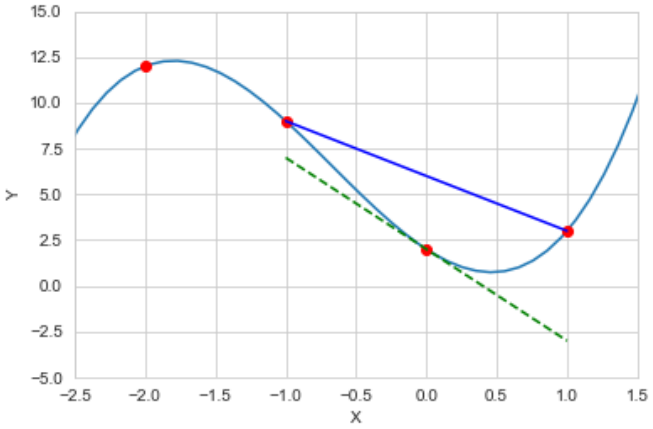
$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$$

kde y_i je hodnota spektra v bodě x_i. Tento vzorec se nazývá centrální diference[8] a jedná se o přesnější aproximaci než jednostranné diference, které mohou být náchylné k chybám na okrajích dat.

Tento vzorec lze zjednodušit pomocí diference $\Delta x = x_{i+1} - x_{i-1}$ a $\Delta y = y_{i+1} - y_{i-1}$:

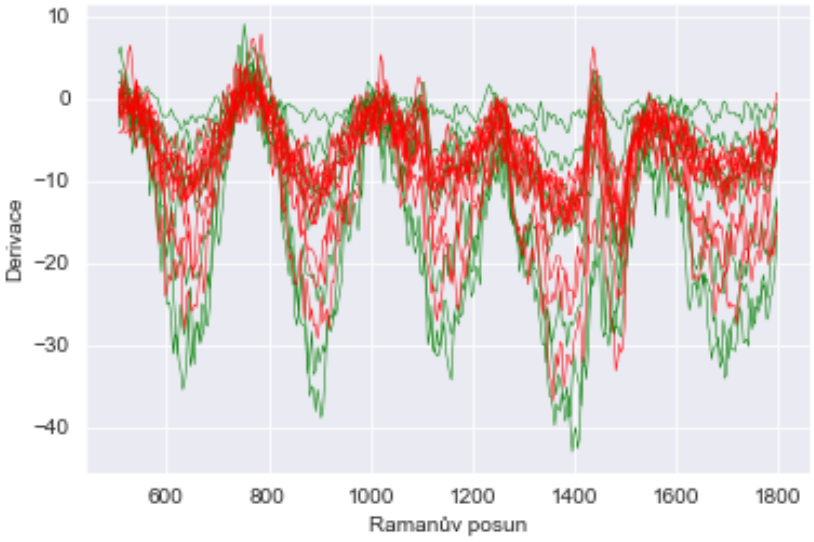
$$y'(x_i) \approx \frac{1}{2}\frac{\Delta y}{\Delta x}$$

Tento vzorec umožňuje vypočítat derivaci pro každý bod spektra na základě sousedních bodů. Tento postup se nazývá numerická diferenciace a je běžnou metodou pro derivování experimentálních dat.



Graf 2.6: Porovnání centrální derivace s derivací

V Grafu 2.6 můžeme vidět čerchovaně skutečnou derivaci a modře centrální derivaci. Je důležité poznamenat, že použití této metody vyžaduje, aby pro každý bod x_i existoval právě jeden bod y_i. Kromě toho mohou vzniknout chyby způsobené šumem dat, což může vést ke zkreslení výsledků derivace.

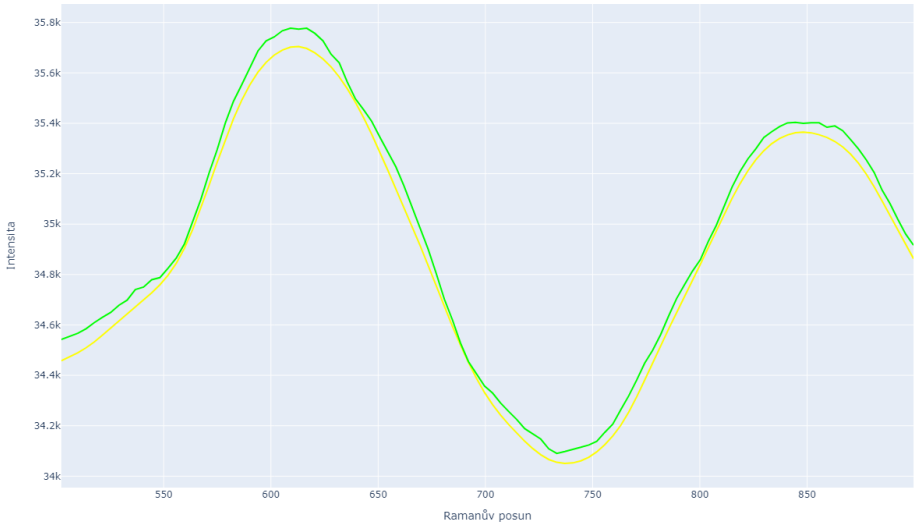


Graf 2.7: Zderivovaná spektra

Po aplikaci derivace vypadají data již vhodně upravená pro následnou klasifikaci. Po konzultaci s odborníkem bylo odhaleno, že velké peaky, které se po transformaci objevily, způsobuje fluorescence. Vliv fluorescence je potřeba odstranit. Je více možností, jak fluorescenci odstranit, jako například MCR[1] nebo EEM[2]. Jedna z možností je také baseline korekce, kterou jsme nakonec použili.

2.3.2 Baseline correction

Mějme vektor y reprezentující měření, které může obsahovat šum a pomalé trendy. Cílem baseline korekce je odstranit pomalé trendy z vektoru y, což nám umožní analyzovat signál v zbytkové části. V Grafu 2.8 můžeme vidět zeleně značené měření a žlutě baseline.



Graf 2.8: Příklad baseline na jednom spektru

Baseline korekce může být provedena pomocí různých metod, jako například

[1]Multivariate Curve Resolution
[2]Excitačně-emisní korekce

metodou polynomu nebo metodou hladké křivky. Zde se zaměříme na metodu algoritmu s minimálními čtverci (ALS-Asymmetric Least Squares Smoothing).[9]

Metoda ALS hledá korekci vztahem

$$\mathbf{y} = \mathbf{b} + \mathbf{s},$$

kde $\mathbf{b}$ je baseline (pomalý trend) a $\mathbf{s}$ jsou špičky (rychlé změny). Cílem je nalézt vektor $\mathbf{b}$, který minimalizuje kvadratickou chybu mezi vektorem $\mathbf{y}$ a jeho dekompozicí na baseline $\mathbf{b}$ a špičky $\mathbf{s}$:

$$\min_{\mathbf{b}} \left(||\mathbf{y} - \mathbf{b} - \mathbf{s}||^2 + \lambda ||\mathbf{D}\mathbf{b}||^2 \right),$$

kde $\mathbf{D}$ je matice diferenčních operátorů[10] a λ je regularizační parametr, který řídí hladkost baseline. V tomto výrazu první člen $||\mathbf{y} - \mathbf{b} - \mathbf{s}||^2$ reprezentuje kvadratickou chybu mezi vektory $\mathbf{y}$ a $\mathbf{b} + \mathbf{s}$ a zajišťuje, že nalezený baseline a špičky co nejlépe odpovídají původnímu signálu $\mathbf{y}$. Druhý člen $\lambda ||\mathbf{D}\mathbf{b}||^2$ reprezentuje regularizaci a zajišťuje, aby byl baseline dostatečně hladký. Matice diferenčních operátorů $\mathbf{D}$ slouží k získání derivací baseline, a to například jako rozdíl mezi sousedními hodnotami vektoru.

Pro minimalizaci výše uvedeného výrazu můžeme použít metodu nejmenších čtverců (LS)[3] nebo metodu gradientního sestupu (GD)[4]. Metoda ALS je alternativou k těmto metodám, která používá iterační optimalizační postup. Tento postup se opakuje, dokud není dosaženo určitého kritéria ukončení (např. maximálního počtu iterací nebo požadované přesnosti).

V každé iteraci se nejprve minimalizuje výraz s baseline $\mathbf{b}$ při konstantním $\mathbf{s}$ a následně se minimalizuje výraz s $\mathbf{s}$ při konstantním $\mathbf{b}$. Tento postup se opakuje střídavě, dokud není dosaženo požadované přesnosti. My jsme baseline

[3] Z anglického least squares method

[4] Z anglického gradient descent

korekci implementovali v programovacím jazyku Python pomocí následující funkce[9]:

```
def baseline_als(y, lam, p, niter=10):
  L =  len(y)
  D = sparse.csc_matrix(np.diff(np.eye(L), 2))
  w = np.ones(L)
  for i in  range(niter):
    W = sparse.spdiags(w, 0, L, L)
    Z = W + lam * D.dot(D.transpose())
    b = spsolve(Z, w*y)
    w = p * (y > b) + (1-p) * (y < b)
  return b
```

Algoritmus 2.1: Baseline korekce

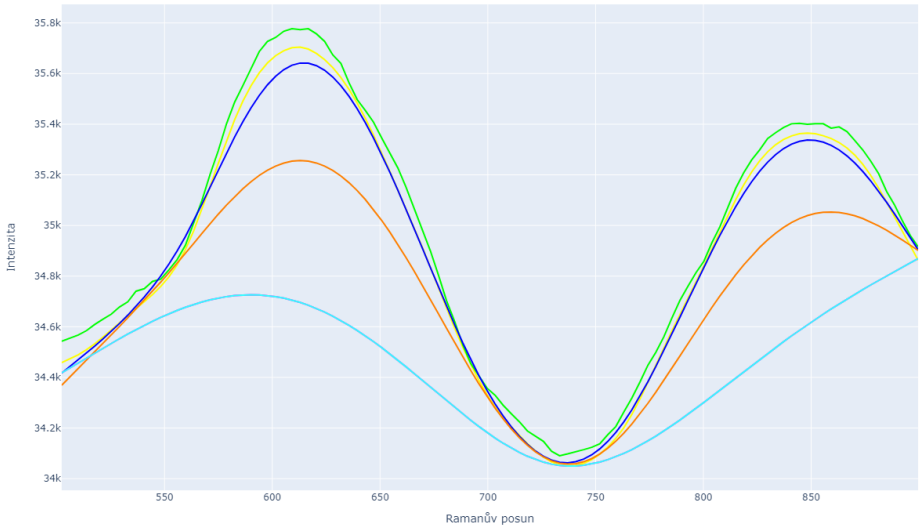
Funkce bere vstupní signál y, parametry lam a p a počet iterací. V průběhu několika iterací algoritmus odhadne základní linii (baseline) signálu pomocí metody nejmenších čtverců, přičemž zahrnuje také adaptivní úpravu vah na základě rozdílu mezi signálem a odhadnutou základní linií. Celkově metoda ALS spočívá v nalezení nejlepší baseline $\mathbf{b}$ a následného odstranění této baseline z původního signálu $\mathbf{y}$ pomocí odečtení vektoru $\mathbf{b}$.

Výsledkem baseline korekce je vektor $\mathbf{y}_{\text{corr}}$, který je získán odečtením nalezené baseline $\mathbf{b}$ od původního signálu $\mathbf{y}$:

$$\mathbf{y}_{\text{corr}} = \mathbf{y} - \mathbf{b}.$$

Výhodou metody ALS je, že dokáže efektivně zpracovat i signály s velkým množstvím šumu a různými typy trendů v datech. Tato metoda je však citlivá na volbu regularizačního parametru λ, který řídí hladkost nalezené baseline $\mathbf{b}$. Pokud je λ příliš velké, baseline bude příliš hladká a mohou chybět některé detaily v dokončeném signálu. Na druhé straně, pokud je λ příliš malé, baseline bude příliš detailní a může obsahovat šum a jiné nežádoucí artefakty.

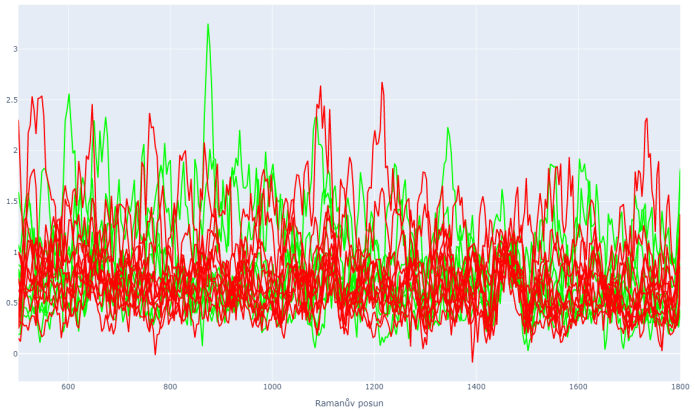
Správná volba hodnoty λ je tedy důležitá pro dosažení nejlepšího výsledku korekce. To můžeme vidět v Grafu 2.9, kde zeleně je značeno měření a zbylé křivky značí baselinu s různým λ.



Graf 2.9: Porovnání parametrů λ

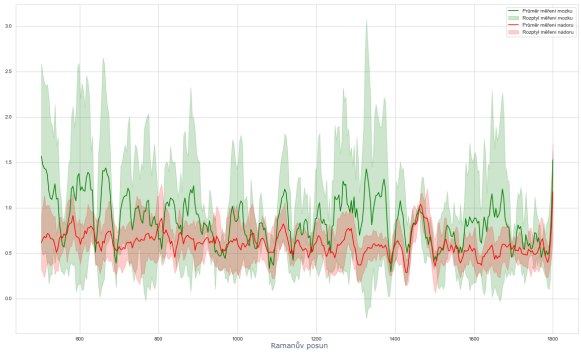
Existuje několik metod, jak určit optimální hodnotu λ. Jednou z nejčastěji používaných metod je křížová validace (cross-validation), při které se data rozdělí na trénovací a validační sady. Poté se provede baseline korekce s různými hodnotami λ na trénovací sadě a výsledné korekce se porovnají s validační sadou. Hodnota λ, která poskytuje nejlepší výsledek na validační sadě, je považována za optimální.

Výsledkem baseline korekce pomocí metody ALS je tedy signál, který je zbaven pomalých trendů a může být dále analyzován s větší přesností a spolehlivostí. Po nalezení vhodného λ můžeme data transformovat odečtením baseliny od původního měření. Výsledek této transformace je vidět v Grafu 2.10.



Graf 2.10: Spektra po odečtení baseline

Mezi měřením nádoru a zdravého mozku by měl být rozdíl v Ramanově posunu $1460,67\,cm^{-1}$. Tato hodnota byla stanovena na základě konzultace s odborníkem. Je proto vhodné data znormovat tak, že každou hodnotu spektra vydělíme hodnotou tohoto spektra v Ramanově posunu $1460,67\,cm^{-1}$. Jelikož zelená spektra jsou v této hodnotě Ramanova posunu obecně níže něž červená, ve výsledku budou zelená spektra v průměru výše než červená. Výsledek takto upravených dat lze vidět v grafu 2.11. Takto upravená data jsou vhodná pro následnou klasifikaci.



Graf 2.11: Průměr s rozptylem spekter

Kapitola 3

Klasifikační metody

V této kapitole se zaměříme na klasifikační metody, které nám umožní kategorizovat data na základě jejich charakteristik. Cílem klasifikačních metod je tedy rozdělit vstupní data na skupiny podle nějakých kritérií. Existuje mnoho různých klasifikačních metod, z nichž každá má své výhody a nevýhody a je vhodná pro určité typy dat. V této kapitole se zaměříme na popis dvou základních klasifikačních metod k-nejbližších sousedů (KNN) a rozhodovacích stromů.

3.1 Měření jako vektor

Předpokládejme, že máme n dat v jednom měření, které bylo upravené normováním a transformacemi. Mějme n hodnot $[x_1, x_2, \ldots, x_n]$. Tyto hodnoty můžeme reprezentovat jako n-dimenzionální vektor

$$y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Tento vektor lze interpretovat jako bod v n-rozměrném prostoru. Každá složka vektoru odpovídá jedné dimenzi tohoto prostoru. Tento přístup má velkou výhodu. Je velmi efektivní z hlediska výpočetního výkonu, protože se jedná o standardní matematické operace s vektory a maticemi.

Vektory lze dále použít jako vstupní data pro klasifikátory. Klasifikátory pak mohou být trénovány na takových datech a přiřazovat je do jedné z

předem definovaných tříd. Z tohoto důvodu se tedy často používá reprezentace dat pomocí vektorů v n-rozměrném prostoru pro účely strojového učení a klasifikace.

3.2 Nejbližší sousedé

Klasifikátor nejbližších sousedů (KNN) je jedním z nejjednodušších a nejčastěji používaných algoritmů pro klasifikaci dat. Jeho základní myšlenkou je, že se nový vzorek přiřadí k třídě nejbližších sousedů v trénovací množině.

Konkrétně předpokládejme, že máme trénovací množinu

$$\mathcal{T} = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n),$$

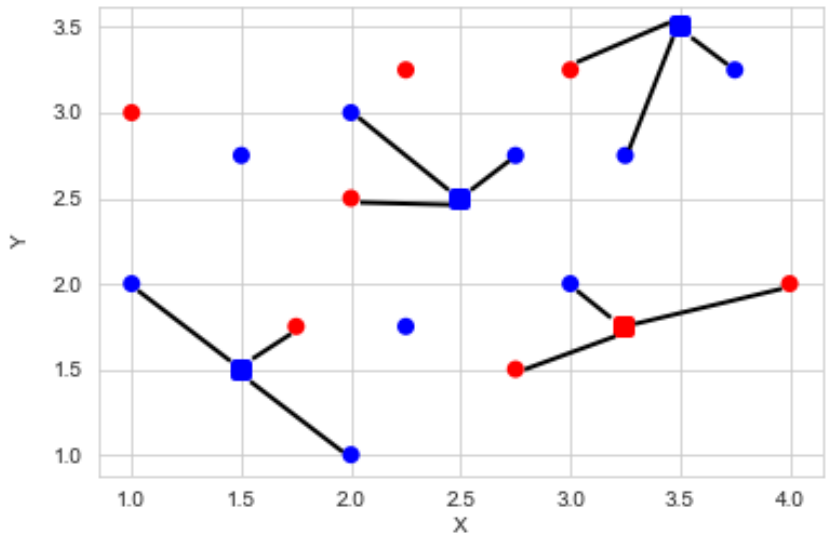
kde x_i jsou vstupní vektory a y_i jsou příslušné třídy. Chceme klasifikovat nový vektor $x_{\text{nový}}$. Algoritmus KNN pracuje tak, že nejprve nalezneme k nejbližších sousedů vektoru $x_{\text{nový}}$ v trénovací množině podle nějaké metriky $d(x, y)$, např. eukleidovské vzdálenosti

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

Poté se vektor $x_{\text{nový}}$ přiřadí do třídy, která se vyskytuje nejčastěji mezi těmito k nejbližšími sousedy. Můžeme ovšem také každému z k nejbližších bodů přiřadit váhu podle vzdálenosti od $x_{\text{nový}}$, ten pak zařadíme do kategorie s vyšším součtem vah.

Algoritmus KNN má několik výhod, jako je jednoduchost implementace, robustnost proti šumu a schopnost pracovat s nelineárními rozhodovacími hranicemi. Nicméně, má také některé nevýhody, například citlivost na volbu metriky a počet nejbližších sousedů.[11]

Za účelem ilustrace tohoto algoritmu použijeme Graf 3.1, který znázorňuje klasifikaci dvoudimenzionálních dat pomocí KNN s $k = 3$:



Graf 3.1: Příklad algoritmu KNN s $k = 3$

Tento graf ukazuje použití KNN pro dvoudimenzionální data. Avšak algoritmus KNN lze použít také pro data vysoké dimenze. Pokud je dimenze dat vysoká, může být obtížné najít nejbližší sousedy, protože se data vysoké dimenze chovají jinak než data nízké dimenze. V takových případech může být užitečné použít některou z metod pro snížení dimenze dat, jako je například PCA[1] nebo t-SNE[2].

3.3 Rozhodovací strom

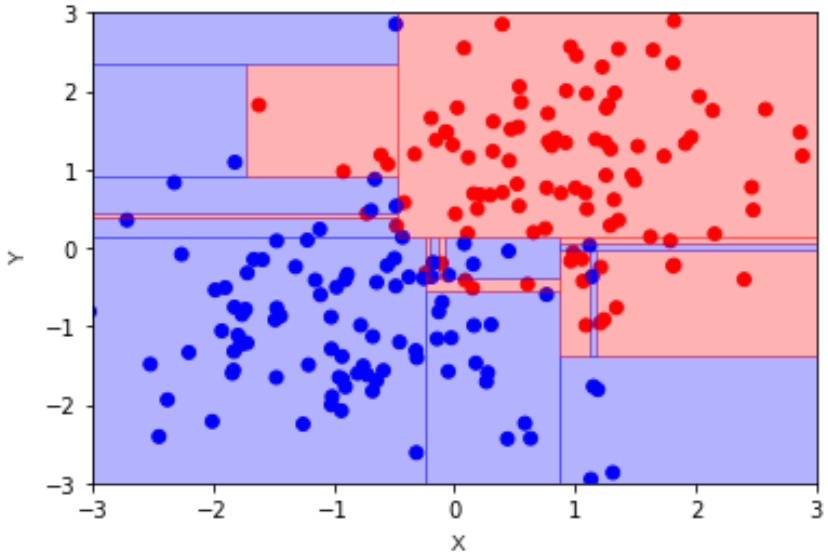
Rozhodovací strom pro vektorová data je matematická metoda třídění vektorů na základě jejich vlastností. V tomto přístupu je vektorový prostor rozdělen pomocí nadrovin, které vytváří hierarchii rozhodování.

[1] Analýza hlavních komponent
[2] t-distributed stochastic neighbor embedding

Každá nadrovina se definuje obecnou rovnicí:

$$a_1 x_i^{(1)} + a_2 x_i^{(2)} + \ldots + a_n x_i^{(n)} = b,$$

kde $x_i^{(j)}$ je j-tá souřadnice vektoru x_i v n-rozměrném vektorovém prostoru a a_i a b jsou konstanty, které určují polohu nadroviny v prostoru. Rozdělení vektorového prostoru na dvě podmnožiny se pak provádí podle toho, na které straně nadroviny se vektor nachází. Koeficienty a_i a b jsou určeny tak, aby rozdělily vektorový prostor tak, aby každá podmnožina obsahovala vektory s podobnými vlastnostmi. Tyto koeficienty jsou získány trénováním algoritmu na trénovacích datech.[12]



Graf 3.2: Příklad rozhodovacího stromu pro dvoudimenzionální data

Výhodou rozhodovacího stromu pro vektorová data je jeho schopnost pracovat s vektorovými daty s různým počtem dimenzí. Rozdělení pomocí nadrovin lze provést pro každou dimenzi zvlášť, což umožňuje efektivně třídit vektory na základě různých vlastností.

Když se vektor dostane do rozhodovacího stromu, postupně projde jednotlivými nadrovinami, dokud není přiřazen do konkrétního listu stromu. Každá nadrovina odpovídá jednomu uzlu stromu a výsledkem je binární strom, kde každý vnitřní uzel má právě dva potomky. Listy stromu představují konečné rozhodnutí o třídě, do které patří daný vektor. Růst rozhodovacího stromu je proces vytváření stromové struktury na základě trénovacích dat. Během růstu stromu se rozhoduje, která vlastnost dat bude použita k rozdělení dat na podskupiny na další úrovni stromu. Proces růstu stromu pokračuje, dokud není dosaženo určitého kritéria pro zastavení. Existuje několik kritérií pro zastavení růstu rozhodovacího stromu.[13] Mezi nejčastější patří:

- Maximální hloubka stromu: Určuje maximální počet úrovní stromu. Pokud je dosaženo této hloubky, růst stromu se zastaví. Čím nižší tento parametr nastavíme, tím jednodušší strom bude.
- Minimální počet vzorků na listu: Určuje minimální počet vzorků na listu stromu. Pokud je počet vzorků menší než tato hodnota, růst stromu se zastaví. Je důležité tento parametr vhodně zvolit, jelikož pokud ho nastavíme příliš nízký, může dojít k $overfittingu$ [3]
- Minimální počet vzorků v uzlu: Určuje minimální počet vzorků v uzlu, které jsou vyžadovány pro rozdělení uzlu na další úroveň v rozhodovacím stromu. I tento parametr musíme volit moudře, protože při příliš vysoké hodnotě dojde k $underfittingu$[4]

[3]Přetrénování rozhodovacího stromu
[4]Nedostatečné natrénování stromu

Kapitola 4

Výsledky

4.1 Křížová validace

Abychom mohli porovnávat výsledky jednotlivých klasifikátorů, je potřeba provést křížovou validaci. Křížová validace (cross-validation) je často používaná metoda pro odhad kvality klasifikátorů. Cílem křížové validace je získat spolehlivý odhad kvality klasifikátoru, který lze porovnávat s jinými klasifikačními modely nebo s očekávanou kvalitou v praxi.

Křížová validace spočívá v rozdělení datové sady na několik menších podmnožin (tzv. "folds"), obvykle stejné velikosti. Každý fold je následně použit jako testovací sada pro model, který byl natrénován na všech ostatních foldech. Takto se vytvoří několik modelů, z nichž každý byl natrénován a otestován na odlišných datech. Výsledkem křížové validace je průměrné skóre modelu na všech testovacích sadách.

Křížová validace je důležitá, protože odhad výkonnosti modelu na jedné testovací sadě může být příliš optimistický nebo pesimistický. Pokud bychom například použili pouze jednu testovací sadu, mohlo by se stát, že náhodně jsou do ní vybrána data, která nejsou příliš reprezentativní pro celou datovou sadu. To by mohlo vést k nadhodnocení nebo podhodnocení výkonnosti.

Křížová validace může být provedena různými způsoby, například pomocí k-fold nebo leave-one-out metody. V případě k-fold metody je datová sada rozdělena na k foldů, a to tak, že každý fold obsahuje přibližně stejný počet vzorků. Poté je provedeno k opakování, přičemž v každém opakování se jeden

fold použije jako testovací sada a ostatní foldy jsou použity k natrénování modelu.[14]

4.2 Jak porovnat výsledky

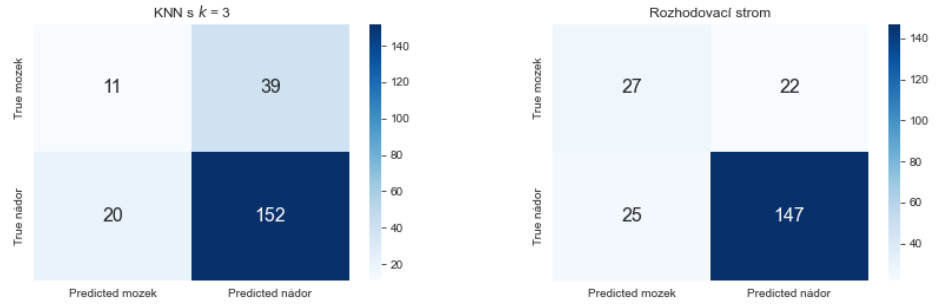
Porovnání výsledků klasifikátoru je důležitou součástí klasifikačních procesů. Existuje několik způsobů, jak hodnotit úspěšnost klasifikátoru. Pro výběr správné metriky je důležité zohlednit celkový kontext a účel klasifikace.

Pokud klasifikujeme pouze do dvou kategorií pro porovnání výsledků klasifikátoru se obvykle používá confusion matrix (matice záměn), která je tabulkou ukazující počty pozitivních, falešných pozitivních, negativních a falešných negativních klasifikací. Na základě této matice je možné vypočítat různé metriky pro hodnocení úspěšnosti klasifikátoru, jako například přesnost (accuracy), úplnost (recall), specifičnost (specificity) a F-míra (F1 score).[15]

V této práci budu používat confusion matrix k porovnání výsledků klasifikátoru, protože je to jednoduchý a efektivní způsob, jak získat ucelený obraz o úspěšnosti klasifikace. Confusion matrix umožňuje snadno porovnávat výsledky různých klasifikátorů a určit, který je nejlepší pro daný účel. Když víme, jak porovnávat výsledky můžeme se pustit na klasifikaci.

4.3 Porovnání výsledků

Když umíme porovnávat výsledky a provádět křížovou validaci, můžeme trénovat a testovat klasifikátory. Výsledky obou klasifikátorů, po provedení křížové validace, můžeme porovnávat pomocí matic záměny, které nám ukážou, jak dobře klasifikátory pracují. Oba klasifikátory byly natrénovány a otestovány na datech po předpracovaní (vyhlazení, normování, transformování).

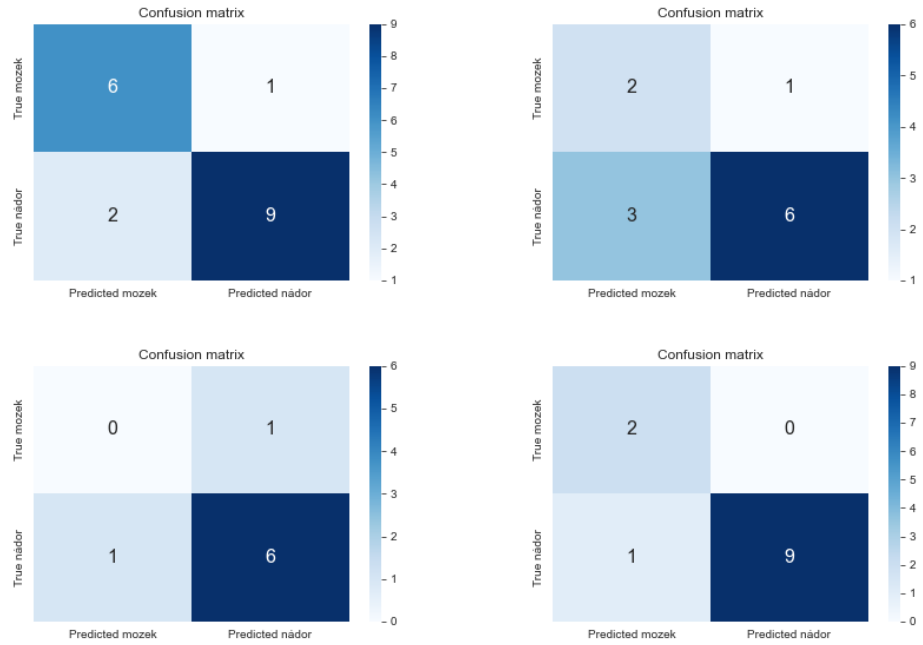


Graf 4.1: Porovnání dvou klasifikátorů

Během porovnávání výsledků klasifikátorů jsme zjistili, že přesnost určení nádoru je u obou klasifikátorů velmi podobná. Naopak při určení zdravého mozku je rozhodovací strom výrazně přesnější. Jedním z důvodů, proč KNN nedosahuje tak dobrých výsledků, může být to, že tento klasifikátor využívá pouze lokální informace a nemá schopnost odhalit globální vztahy mezi daty.

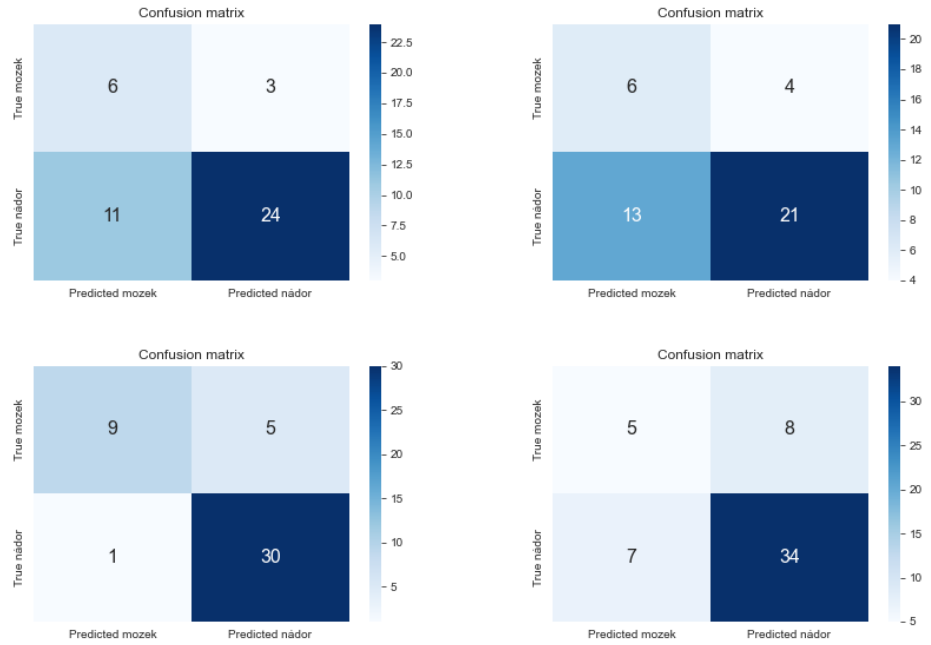
Na druhé straně, rozhodovací strom dosahoval lepších výsledků, a to díky tomu, že umožňuje odhalit globální vztahy mezi daty. Nicméně, při použití rozhodovacího stromu bylo potřeba vhodně nastavit parametry zastavení růstu, aby dosáhl nejlepších výsledků. Bez správného nastavení těchto parametrů by se rozhodovací strom mohl přeučit a výsledky by mohly být horší.

Pokud budeme chtít ověřit, jak dobře bude klasifikátor fungovat v praxi, můžeme použít metodu testování na nových, neznámých datech. Jelikož nová data nemáme, můžeme odebrat spektra 4 pacientů. Na zbylých spektrech klasifikátor natrénujeme a poté mu vložíme spektra odebraných pacientů jako testovací. Tento postup vyzkoušíme pouze na rozhodovacím stromu, jelikož dosahoval značně lepších výsledků.



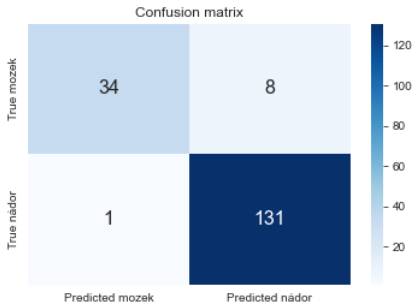
Graf 4.2: Výsledky na jednotlivých pacientech

Jak můžeme vidět z Grafu 4.2, náš klasifikátor v mnoha případech určil správné kategorie. Je ale možné, že jsme pouze šťastně rozdělili pacienty na trénovací a testovací podmnožiny. Abychom tomu předešli, můžeme opět použít křížovou validaci. V tomto případě ale nebudeme rozdělovat všechna spektra do k skupin ale pacienty. Přesněji vybereme náhodně 4 pacienty, na kterých budeme model testovat, a zbylých 15 použijeme jako trénovací.



Graf 4.3: Výsledky po křížové validaci

Očekávali jsme, že výsledky klasifikace nádorových spekter budou vysoké, jelikož si u nich patolog byl jistý. Naopak spektra zdravého mozku, jak nám sdělil odborník, mohou obsahovat nádor, který nebyl patologem detekován. Z tohoto důvodu, může být špatná klasifikace mozku ve skutečnosti správná.



Graf 4.4: Výsledky na trénovací sadě

Jak si můžeme všimnou, výsledky našeho klasifikátoru na množině dat, na

které jsme klasifikátor natrénovali jsou u nádoru téměř stoprocentní. Přesnost určení zdravého mozku není už tak vysoká, ale jak již bylo řečeno, skutečná přesnost může být mnohem vyšší. Pro využití tohoto klasifikátoru v praxi je ale stále příliš nepřesný. Tato nepřesnost může být způsobená nedostatkem trénovacích dat nebo také nevyváženým poměrem mozkových spekter k nádorovým. Pro zlepšení přesnosti klasifikátoru můžeme zvážit několik možností. Jednou z nich je získání více trénovacích dat, zejména pak spekter zdravého mozku.

Je také důležité brát v úvahu kontext, ve kterém bude klasifikátor použit, a přizpůsobit jeho parametry a vlastnosti tomuto kontextu. Například, pokud bude klasifikátor použit pro screening na klinice, může být důležité maximalizovat schopnost detekovat nádor, i když to může vést ke snížení celkové přesnosti. Další možností pro zlepšení přesnosti klasifikátoru je použití pokročilejších algoritmů strojového učení, jako jsou neuronové sítě.

Závěr

V první části této práce jsme se zaměřili na několik metod transformace a normalizace dat z Ramanovy spektroskopie. Pro dosažení vysoké kvality klasifikace je totiž klíčové zajistit, aby data byla správně připravena a zpracována. V této části jsme ukázali, jak lze využít například Savitzky-Golay filtr, derivace či baseline korekci.

V druhé části jsme se zaměřili na základní klasifikátory, jako jsou KNN nebo rozhodovací strom a vysvětlili jejich principy, výhody a nevýhody. Poté jsem tyto klasifikátory použili k samotné klasifikaci dat z Ramanovy spektroskopie mozkové tkáně. Výsledky byly slibné a ukázaly, že tato technologie může být velmi užitečná pro diagnostiku a výzkum v oblasti neurovědy.

Celkově tedy tato práce ukazuje, jak může být tato technologie využita k dosažení relativně vysoké kvality klasifikace dat a tím přispět ke zlepšení diagnostiky a léčby v oblasti neurovědy. Práce ukazuje nejlepší možnost, jak by se mohla data z Ramanovy spektroskopie transformovat a následně klasifikovat, vzhledem k dodaným datům. Bohužel, kvůli nedostatku dat je zatím obtížné dospět k definitivnímu závěru. Nicméně v práci na této problematice se bude pokračovat dále a budeme se snažit vytvořit co nejlepší možný klasifikátor s využitím většího množství dat a odhalit znaky, na základě kterých se bude klasifikátor rozhodovat pro možnost následné analýzy.

Literatura

1. LACROIX, Michel; BI-SAID, Dany; FOURNEY, Daryl R; AL., et. A multivariate analysis of 416 patients with glioblastoma multiforme: prognosis, extent of resection, and survival. *Journal of Neurosurgery*. 2001, roč. 95, č. 2, s. 190–198.
2. VAQAS, Babar; SMITH, Richard JH; HALL, Gregory; AL., et. Raman spectroscopy: a novel tool for intraoperative guidance in surgical neuro-oncology. *Neuro-Oncology*. 2018, roč. 20, č. Suppl 2, s. ii16.
3. DIEM, Max. *Modern Vibrational Spectroscopy and Micro-Spectroscopy*. Chichester, UK: John Wiley & Sons, 2015. ISBN 9781118824924.
4. NOVÁKOVÁ, Miroslava. *Implementace Savitzky-Golay filtru pro zpracování biologických signálů*. Ostrava, 2021. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava.
5. RAIDA, Zbyněk. *Metoda nejmenších čtverců pro odhad parametrů*. 2001. Dostupné také z: https://www.radio.feec.vutbr.cz/raida/optimalizace/squares/squares_a.htm.
6. PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand; GRISEL, Olivier; BLONDEL, Mathieu; PRETTENHOFER, Peter; WEISS, Ron; DUBOURG, Vincent; VANDERPLAS, Jake; PASSOS, Alexandre; COURNAPEAU, David; BRUCHER, Matthieu; PERROT, Matthieu; DUCHESNAY, Edouard. *MinMaxScaler* [https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html]. 2021. [Online; accessed 6-April-2023].
7. DEVELOPERS, Scikit learn. *Standardization, or mean removal and variance scaling* [Scikit-learn 0.24.1 documentation]. 2018. [cit. 2021-04-10]. Dostupné z: https://scikit-learn.org/stable/modules/

preprocessing.html#standardization-or-mean-removal-and-variance-scaling.

8. MÁLEK, Josef; HECZKO, Jan. *Numerická derivace a integrace* [Masarykova univerzita]. 2020. Dostupné také z: https://is.muni.cz/el/sci/jaro2020/C1471/um/numderint.pdf.

9. EILERS, Paul; BOELENS, Hans. Baseline Correction with Asymmetric Least Squares Smoothing. *Unpubl. Manuscr.* Listopad 2005.

10. SHCHUR, Andriy. *The Matrix of Differentiation* [https://www.math.drexel.edu/~tolya/matrix]. 2017. Accessed: 2021-04-10.

11. DEVELOPERS, Scikit learn. *sklearn.neighbors.KNeighborsClassifier* [https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html]. 2021. [Accessed: 2022-01-06].

12. KOMPRDOVÁ, Klára. *Rozhodovací stromy a lesy*. Plzeň: Nakladatelství ZČU v Plzni, 2012. ISBN 978-80-7204-785-7.

13. KASHYAP, Gaurav. 3 Techniques to Avoid Overfitting of Decision Trees. *Towards Data Science* [https://towardsdatascience.com/3-techniques-to-avoid-overfitting-of-decision-trees-1e7d3d985a09]. 2021. [Online; accessed 6-April-2023].

14. BROWNLEE, Jason. K-Fold Cross Validation. *Machine Learning Mastery* [https://machinelearningmastery.com/k-fold-cross-validation/]. 2018. [Online; accessed 6-April-2023].

15. DEVELOPERS, Scikit learn. *Model Evaluation: Quantifying the Quality of Predictions*. 2021. Dostupné také z: https://scikit-learn.org/stable/modules/model_evaluation.html. Accessed: 2021-04-11.

Seznam grafů