VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

NÁSTROJ PRO PREDIKCI ATRIBUTŮ ŽIVOTNÍHO STYLU NA ZÁKLADĚ METAGENOMICKÝCH DAT Z TLUSTÉHO STŘEVA

TOOL FOR CLASSIFICATION OF LIFESTYLE TRAITS BASED ON METAGENOMIC DATA FROM THE LARGE INTESTINE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN KUBICA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. STANISLAV SMATANA**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Kubica Jan**
Program: Informační technologie
Název: **Nástroj pro predikci atributů životního stylu na základě metagenomických dat z tlustého střeva**
Tool for Classification of Lifestyle Traits Based on Metagenomic Data from the Large Intestine
Kategorie: Bioinformatika

Zadání:

1. Seznamte se se základními principy metagenomiky a s jejím využitím při analýze mikrobiomu lidského střeva pomocí amplikonového sekvenování 16S rRNA.
2. Seznamte se s existujícími přístupy pro zpracování sekvenačních dat typu 16S rRNA a s existujícími metodami klasifikace fenotypových rysů na základě těchto dat.
3. Navrhněte nástroj na klasifikaci vybraných atributů životního stylu jedinců (masožravec/vegetarián, kuřák/nekuřák, atd...) na základě složení jejich střevního mikrobiomu.
4. Navržený nástroj implementujte a jeho činnost ověřte na vhodně vybraných testovacích vzorcích. Experimentujte s ruznými typy klasifikátorů a různými nastaveními parametrů.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího pokračování projektu.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Smatana Stanislav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 26. října 2018

Abstrakt

Tato práce se zabývá analýzou lidského mikrobiomu na základě metagenomických dat z tlustého střeva. Předmětem zkoumání je zastoupení bakterií na různých taxonomických úrovních v závislosti na životním stylu jedince. Byl vytvořen nástroj klasifikující jednotlivé atributy, jako jsou stravovací návyky (vegetarián, vegan, všežravec), citlivost na lepek a laktózu, body mass index nebo věk či pohlaví, s využitím metod strojového učení. Při implementaci byly zvoleny metody k nejbližších sousedů (kNN), náhodný les (RF) a metoda podpůrných vektorů (SVM). Data pro natrénování klasifikátoru a vyhodnocení byla čerpána z projektu American Gut. Práce se rovněž zaobírá problémy spojenými s danými datovými sadami, jako je mnoharozměrnost, řídkost, jejich kompoziční závislost a nevyváženost.

Abstract

This thesis deals with analysis of human microbiome using metagenomic data from large intestine. The main focus is placed on bacteria composition in a sample on different taxonomic levels regarding the lifestyle traits of an individual. For this purpose, a tool for classification of several attributes was created. It considers attributes like diet type and eating habits (vegetarian, vegan, omnivore), gluten and lactose intolerance, body mass index, age or sex. From range of machine learning perspectives considering K Nearest Neighbours (kNN), Random Forest (RF) and Support Vector Machines (SVM) were used. Datasets for training and final evaluation of the classifier were taken from American Gut project. The thesis also focuses on particular problems with metagenomic datasets like its multidimensionality, sparsity, compositional character and class imbalance.

Klíčová slova

metagenomika, taxonomie, OTU, predikce, klasifikace, strojové učení, k nejbližších sousedů, metoda podpůrných vektorů, náhodný les, T-test, analýza hlavních komponent, lineární diskriminační analýza

Keywords

metagenomics, taxonomy, OTU, prediction, classification, machine learning, K Nearest Neighbours, Support Vector Machines, Random Forest, T-test, Principal Component Analysis, Linear Discriminant Analysis

Citace

KUBICA, Jan. *Nástroj pro predikci atributů životního stylu na základě metagenomických dat z tlustého střeva.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Stanislav Smatana

Nástroj pro predikci atributů životního stylu na základě metagenomických dat z tlustého střeva

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Stanislava Smatany. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Kubica
22. května 2019

Poděkování

Rád bych poděkoval Ing. Stanislavovi Smatanovi za jeho vedení, trpělivost a odbornou pomoc během tvorby této práce. Zároveň bych chtěl poděkovat všem, kteří mě v průběhu mnohokrát jakkoliv podpořili.

Obsah

Kapitola 1

Úvod

Mikroorganismy se nachází všude kolem nás. I zdravé lidské tělo je pro mnohé z nich typickým domovem. Převážnou část těchto mikrobů tvoří bakterie, které se mohou vyskytovat téměř kdekoliv na pokožce, v ústech, v okolí intimních partií nebo ve větší míře právě v našem trávicím traktu a tlustém střevě jako součást naší střevní mikroflóry. Takovéto společenství mikroorganismů nazýváme jako mikrobiom [26].

Obvykle tato společenstva zajišťují široké množství biochemických dějů a mají přímý vliv na celkové zdraví daného jedince, umožňují extrahovat živiny a energii ze stravy nebo přispívají ke správné funkci imunitního systému [12]. Nerovnováha v zastoupení střevního mikrobiomu mít podle některých odborných článků spojitost s obezitou a potravinovými alergiemi [8], chronickými onemocněními střev či jinými trávicími obtížemi [4].

Až donedávna bylo zkoumání celkového složení lidského mikrobiomu značně omezené, neboť za pomoci klasických metod, jako je kultivace daného vzorku, většina druhů bakterií umírá a jejich diverzita se snižuje [12]. V současnosti existují i odlišné postupy. Tyto postupy staví na zkoumání veškeré genetické informace získané ze vzorku, které říkáme metagenom [20]. S rozvojem metod sekvenování a současně klesající ceně těchto úkonů, přibývá množství metagenomických dat ve veřejně dostupných databázích. Tato data dále vybízí k podrobnějším analýzám.

Každý člověk má svou vlastní bakteriální stopu a u každého jedince se zastoupení bakterií mírně liší. Částečné odlišnosti mohou být sice způsobeny sběrem vzorků lidí různých národností, vyskytujících se vzájemně v odlišném prostředí, mohou ale taktéž být poměrně dobrými ukazateli a odrážet životní styl, věk a jiné charakteristiky [12][4]. S nadsázkou by se dalo říci, že náš vlastní metagenom o nás dokáže leccos prozradit.

Cílem této práce je vytvořit nástroj, který by dokázal predikovat jednotlivé životní atributy podle informací získaných z metagenomických dat. Na začátku práce je čtenář seznámen se základními prvky metagenomiky (2) a s využívanými přístupy analýzy. Třetí kapitola (3) je zasvěcena základním principům strojového učení a popisuje klasifikační algoritmy, které by se mohly ukazovat jako vhodné pro daná metagenomická data. Čtvrtá kapitola (4) popisuje teorii, která je potřebná k řešení problémů spojených datovou sadou a byla dostudována v průběhu analýzy výsledků implementovaného nástroje. Daný nástroj byl tak následně o tyto poznatky implementačně rozšířen.

S pátou kapitolou (5) započíná část popisující praktickou tvorbu nástroje. Nejdříve je popsáno vytvoření vhodné datové sady, která by mohla posloužit jako trénovací sada pro budoucí klasifikátor. Následující šestá kapitola (6) popisuje základní návrh nástroje pro predikci. Sedmá kapitola (7) objasňuje implementační detaily nástroje a je v úzké sou-

vislosti s osmou kapitolou (8), kdy reaguje na výsledky získané v průběhu práce na daném nástroji. V závěru (9) jsou zhodnoceny dosažené výsledky a diskutovány možnosti dalšího pokračování projektu.

Kapitola 2

Základy metagenomiky

2.1 Metagenomika

Metagenomika je vědní obor zabývající studiem genetického materiálu získaného přímo ze vzorku a jeho biologickou diverzitou. Představuje soubor molekulárně biologických metod, které pracují s veškerou genetickou informací dostupnou v daném mikrobiálním společenství a poskytuje tak daleko větší množství informací o biodiverzitě než postupy založené na kultivaci [20]. Metagenomika je alternativou ke klinické a enviromentální mikrobiologii. V případě klinické a enviromentální mikrobiologie je daný vzorek nejprve po určitý čas kultivován, dokud se bakterie nerozmnoží v dostatečné míře, a poté je vyhodnocen. Během tohoto procesu ale běžně značná část bakterií ve vzorku zahyne, neboť v laboratorních podmínkách není možné přesně simulovat jejich přirozené prostředí [20]. V případě metagenomiky není kultivace potřeba a daný vzorek je tak možné analyzovat daleko dříve.

2.2 Přístupy k sekvenování

Ve spojení s metagenomikou je využíváno dvou přístupů sekvenování [12]:

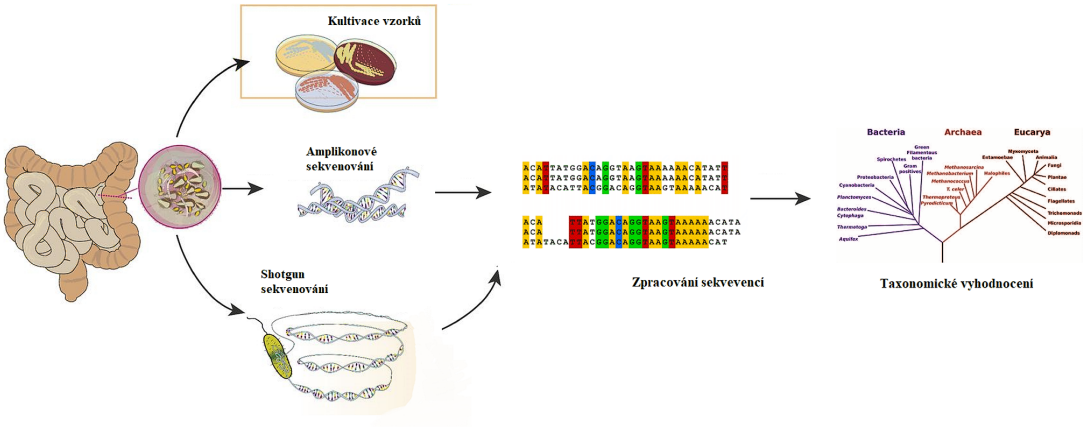
- **amplikonová sekvenace**
- **shotgun sekvenace**

Metody založené na *amplikonovém sekvenování* spočívají v namnožení *(amplifikaci)* pouze specifické části DNA, která obsahuje dva druhy úseků, a to úseky velmi proměnlivé *(hypervariabilní)* a stálé *(konzervativní)*. Zatímco stálé úseky slouží k snadné identifikaci dané části DNA pro amplifikaci a současně jsou společné pro většinu druhů bakterií, proměnlivé úseky se mezi jednotlivými druhy liší. Na základě těchto proměnlivých úseků pak vznikají referenční databáze umožňující přiřazení těchto úseků ke konkrétním taxonomickým druhům. Amplikonová sekvenace je dnes nejběžněji využívanou metodou při studiu bakteriální části lidského mikrobiomu a naprostá většina dostupných metagenomických dat je tohoto typu.

Alternativou k amplikonové sekvenaci je *shotgun sekvenace*. Ta je založena na zkoumání veškeré izolované DNA z odebraného vzorku, kde se nacházejí různě dlouhé úseky *DNA* každé bakterie. Tato metoda je sice vhodnější a přesnější při zkoumání taxonomického zastoupení organizmů, protože zde není zapotřebí žádné amplifikace. Na druhou stranu

vzhledem k nutnosti sekvenování velkého množství materiálu *DNA*, je však časově i finančně náročná a z tohoto důvodu tento postup není tak častý.

Na obrázku 2.1 můžeme vidět jak klasickou metodu kultivace, tak dvě rozdílné metody pro získání genetické informace, počítačové zpracování získaných sekvencí a nakonec vyhodnocení zastoupení jednotlivých bakterií ze vzorku.



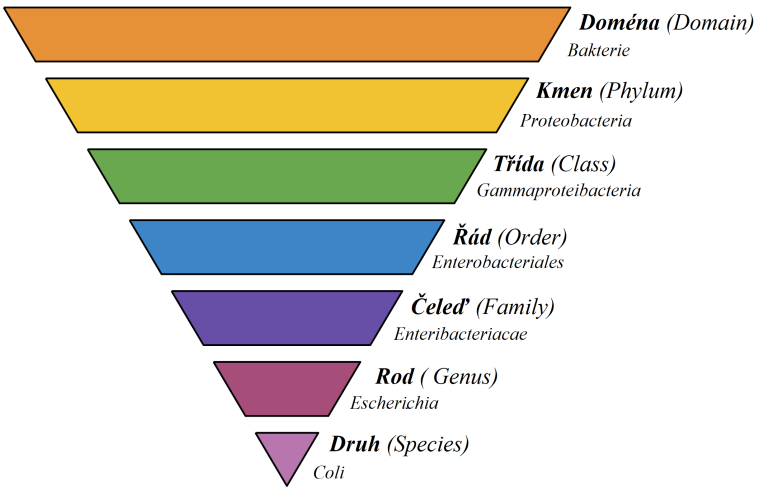
Obrázek 2.1: Přístup analýzy mikrobiálních společenstev v tlustém střevě [12][20]

2.3 Taxonomické rozdělení

V rámci systematické biologie je každý živý organismus klasifikován dle společných znaků do konkrétní kategorie organismů, kterou označujeme jako *taxon*. Každý jednotlivý taxon je pak možné podle určitých kritérií zařadit do jedné společné hierarchické struktury, kterou nazýváme jako *strom života* nebo taktéž *fylogenetický strom*. Pro zařazení bakteriálních společenstev se běžně se využívá sedmiúrovňový systém, jehož stupně jsou:

- **Doména** – *Domain*
- **Kmen** – *Phylum*
- **Třída** – *Class*
- **Řád** – *Order*
- **Čeleď** – *Family*
- **Rod** – *Genus*
- **Druh** – *Species*

V následujících kapitolách jsou pro zjednodušení tyto úrovně označované písmenem *„L“* a číslem od *2* po *6*, reprezentující postupně kmen, třídu, řád, čeleď a rod. Úroveň *L1* je při vyhodnocení zanedbána, tato úroveň rozděluje mikroorganismy pouze na dvě části, a to bakterie a archea. Mikroorganismy z domény archea se v tlustém střevě téměř nenachází. Zařazení mikroorganismů až do hloubky *L7* není ve většině případů metagenomických analýz vyhodnocováno.

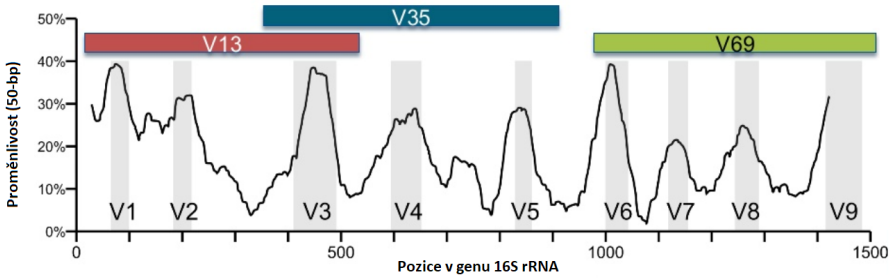


Obrázek 2.2: Příklad taxonomického zařazení bakterie Escherichia Coli [24]

2.4 16S rRNA

Pro účely amplikonové sekvenace popsané v sekci 2.2 se jako nejvhodnější ukazuje využít ribozomální *RNA (rRNA)*, nebo přesněji pouze její konkrétní část, kterou označujeme jako *16S rRNA*. Tento segment *16S rRNA* má adekvátní zastoupení stálých a proměnlivých regionů, kde dochází daleko rychleji k jednotlivým mutacím. Proměnlivé regiony jsou znázorněné v grafu 2.3 jako *V1* až *V9*.

RNA je však nestabilní molekula a její analýza je technicky náročná. Běžně se tak nesekvenuje *rRNA*, ale gen *DNA*, který tuto *rRNA* kóduje [5]. Celý bakteriální gen *16S* se skládá ze zhruba 1500 nukleových bází *(nukleotidů)*, značených písmeny *A*, *T*, *G* a *C*. Každé z těchto písmen reprezentuje konkrétní nukleotid - *Adenin*, *Thymin*, *Guanin* nebo *Cytosin*. Podle různé kombinace těchto písmen je pak možné daný organismus taxonomicky klasifikovat.



Obrázek 2.3: Graf ukazující rozdílnou proměnlivost mezi jednotlivými úseky genu 16S [13]

2.5 Řetězcové sekvence

Současné sekvenátory nové generace produkují velké množství kratších úseků *DNA (tzv. „readů“)*, což představuje problém pro počítačové zpracování. Náročnost skládání je pak dána délkou a počtem fragmentů, který je rozdílný podle *hloubky sekvenování*. Většinou jsou tyto sekvence současně doplněny informací o kvalitě čtení jednotlivých nukleových kyselin. Zpracování surových metagenomických dat je poměrně složitý a výpočetně náročný úkol, který je zpracováván na výpočetních serverech [10]. Formát pro řetězcové sekvence nabírá buď přípony *.fasta*, nebo *.fastq*, pokud jsou jednotlivé čtení doplněny o informaci o kvalitě čtení každého nukleotidu.

```
@ERR1073439.1 10317.000001001_0/1
TACGGAGGGTGCAAGCGTTAATCGGAATCACTGGGCGTAAAGCGCACGTAGGCGGCCTGGTAAGTCAGGGGTGAAATCCCACAGCCCAACTGTGGAACTGCCTTTGA
+
CCCCCCCBBBCCGGGGGGGGGGHGGGGGHHHHHHGHGGGGHHHGGGGGHHHGHGGGGGGHHHHHHHHHGHHGGGHHHHHHHHHGHGGGHHGHHHHHHHHHHHHHHHH
@ERR1073439.2 10317.000001001_1/1
TACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCACGCAGGCGGTCTGTCAAGTCGGATGTGAAATCCCCGGGCTCAACCTGGGAACTGCATTCGA
+
CCCCCCCCCBABGGGGGGGGGGHGGGGGHHHHHGHHGGGGHGHGGGGGGGGGGGGGGGHHHHHGHHGGEGGHHHHHHHHHGGGGEGHHHHHHHHGHHGHHHHHHHGH
@ERR1073439.3 10317.000001001_2/1
TACGTAGGTGGCGAGCGTTATCCGGAATTATTGGGCGTAAAGAGGGAGCAGGCGGCACTAAGGGTCTGTGGTGAAAGATCGAAGCTTAACTTCGGTAAGCCATGGAA
+
BBBBBFFB5DAFAEGGGGGGGGHGGGCGHH5FFFGFGGGGHHBGEFEEFHH1EEG/@E?33@3GHHGHFFHGFHHHH33FGG3FEGHHHHBEFFGFHFHFHHHHHGH
@ERR1073439.4 10317.000001001_3/1
TACAGAGGTCTCAAGCGTTGTTCGGAATCACTGGGCGTAAAGCGTGCGTAGGCTGTTTCGTAAGTCGTGTGTGAAAGGCGCGGGCTCAACCCGCGGACGGCACATGA
+
BBBBBFFBFFFFGGGGGEGCGGHGGGGGHHHHHHHHGEGGHHHEGGGGGGGGHHHHHHHGHFHHHHGGEHHHHFHHHGHGGGGGECGHGHHGGGGGGGGGGGHHHHH
```

Obrázek 2.4: Příklad formátování FASTQ souboru s informací o kvalitě

2.6 Operační taxonomická jednotka OTU

Jelikož sekvenační stroje produkují velké množství sekvencí (řádově desítky až stovky tisíc), a současně jsou tyto čtení díky amplifikaci znásobena, obvykle se přistupuje ke sdružení vzájemě podobných sekvencí. Tento proces je označován jako *OTU picking* a má za následek značné usnadnění dalšího zpracování. Velmi často je původní množství sekvencí (řádově miliony) zredukováno na cca tisícovky clusterů. Operační taxanomickou jednotkou *(OTU - Operational taxanomic unit)* je tedy označován shluk *(cluster)* sekvencí s předem stanvenou podobností [13]. Obvykle je mezi odborníky používána podobnost s hranicí 97%, avšak je možné klasifikovat na vyšším prahu podobnosti jako např. 99%.

2.7 RDP klasifikátor

RDP klasifikátor *(Ribosomal Database Project Classifier)* je klasifikátor, který přiřazuje jednotlivé taxony k shlukovaným úsekům DNA podle referenčních sekvencí. Tyto klasifikátory úzce souvisí s dostupnou referenční databází. Při nejčastějším způsobu shlukování jsou shluky tvořeny na základě *97%* podobnosti. V takovém případě je možné ke shlukům přiřadit taxonomické zařazení až do úrovně *L6*. Při shlukování na jiných mírách podobnosti jsou shlukům přiřazeny taxony jiné úrovně. Například pro *99%* podobnost je snaha vyhodnotit vzorky až po úroveň *L7 (genus)*, při podobnosti 90% jsou shluky rozděleny po taxony na úrovni *L5 (family)* a při nižší podobnosti např. 85% je shlukům přiřazena pouze třída *(L3 - class)*. [3] Mezi v současnosti používané referenční databáze patří databáze *Greengenes*, *SILVA*, *NCBI* a jiné.

2.8 Tabulka OTU

Po úspěšném procesu shlukování sekvencí jsou tato data uspořádána do tabulky, které říkáme *tabulka OTU*. Tato tabulka vyjadřuje počet jednotek *OTU* každého vzorku přiřazených pomocí *RDP* klasifikátoru k různým taxonů. *OTU* tabulka se tedy skládá ze dvou os, kdy na jedné ose jsou vyneseny analyzované vzorky a na druhé identifikátory shluků *(taxony)* podle referenční databáze. Zastoupení jednotek, které byly přiřazeny konkrétnímu taxonu, lze následně použít jako rysy pro klasifikátory, které zakládají na algoritmech strojového učení. Standardní formát pro tabulky *OTU* je formát *BIOM*. Tabulka může nabývat i jiných formátů, jako např. *CSV* pro účely této práce. Příkladem může být obrázek 2.5 popisující tabulku *OTU* pro taxonomickou úroveň *L2*.

	A	B	C	D	E
1	ID	k__Bacteria;p__Firmicutes	k__Bacteria;p__Proteobacteria	k__Bacteria;p__Bacteroidetes	k__Bacteria;p__Actinobacteria
2	ERR1077551	4662	5203	899	1482
3	ERR1072624	960	12192	2886	16
4	ERR1072625	3570	6535	7204	6
5	ERR1072626	862	15609	5331	7
6	ERR1072627	248	7108	486	23
7	ERR1072628	9083	7502	5497	17
8	ERR1072629	2797	3989	14432	16
9	ERR1072630	5848	1538	16181	44
10	ERR1072631	17044	8520	6816	882
11	ERR1072632	3328	14659	3688	14
12	ERR1072633	14481	1799	20918	2182
13	ERR1072634	4607	14321	3668	37
14	ERR1072635	19349	183	4468	160

Obrázek 2.5: Znázornení tabulky OTU

Kapitola 3

Základy strojového učení

V této kapitole si stručně popíšeme, jakým způsobem je možno predikovat životní atributy jedince, které jsou předmětem zkoumání této práce. Postupně se seznámíme s základními principy strojového učení, představíme si algoritmy strojového učení použité během implementační části této práce, metody pro zvýšení efektivity strojového učení, přístupy testování a možnosti vyhodnocení s výčtem několika metrik.

Životní atributy můžeme rozdělit podle charakteru hodnot na dva typy:

- **Kvalitativní hodnoty** - diskrétní veličiny možné vyjádřit výčtem hodnot
- **Kvantitativní hodnoty** - spojité veličiny v rozmezí intervalu

Strojové učení umožňuje predikovat hodnoty obojího typu, predikci kvalitativních hodnot nazýváme *klasifikace*, u kvantitativních hodnot pak mluvíme o *regresi*. Algoritmy zmíněné v následujících sekcích mohou být primárně určeny pro klasifikaci, jejich modifikace ale často umí pracovat i s hodnotami kvantitativními.

Dané hodnoty predikujeme vždy vzhledem k jednotlivým vlastnostem každého vzorku. Tyto vlastnosti označujeme jako *rysy*. Rysem může být jakákoliv informace dostupná o daném vzorku, nejčastěji se jedná však o číselnou reprezentaci určité veličiny jako je délka nebo váha, v našem případě množství jednotek OTU.

3.1 Typy strojového učení

Strojové učení můžeme podle způsobu učení rozdělit do několika kategorií:

1. **Učení s učitelem** představuje situaci, kdy pro vstupní data existuje explicitně správný výstup, resp. třída pro klasifikaci či hodnota pro regresi.

2. **Učení bez učitele** je případ, kdy daný algoritmus musí najít určité vzory, avšak ke vstupním datům neexistuje očekávaný výstup.

3. **Zpětnovazební učení** je učení, kdy správný výstup není poskytován od učitele, ale hodnocení je odvozeno na základě prostředí.

Metody využité v této práci pracují na základě strojového učení s učitelem.

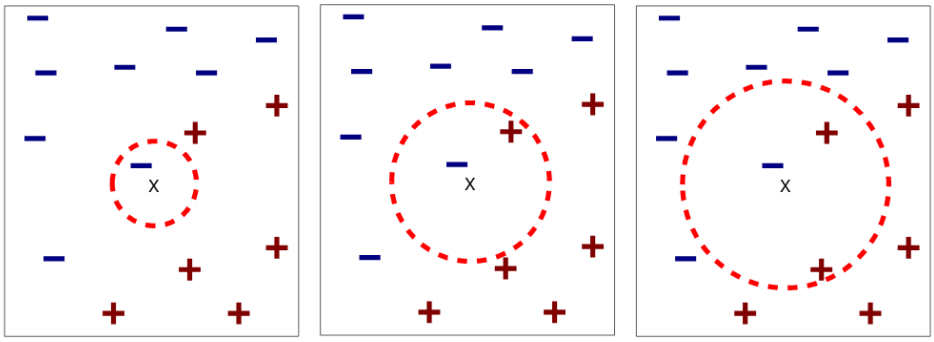
3.2 Klasifikační algoritmy

V této části si představíme tři metody strojového učení, podle kterých je možno data klasifikovat a za pomoci různých modifikací umožňují tyto algoritmy taktéž řešit i regresní problémy. Všechny metody byly vyzkoušeny během implementace. Informace k daným algoritmům jsou čerpané ze zdrojů [7], [11] a [9].

3.2.1 K nejbližších sousedů - kNN

K nejbližších sousedů *(K-Nearest Neighbours)* je základní algoritmus strojového učení založený na podobnosti s jinými prvky v prostoru, které nazýváme nejbližšími sousedy. Princip algoritmu vychází z předpokladu, že cíle, které chceme predikovat, budou mít podobné vlastnosti (hodnoty atributů) jako některá data z trénovací množiny, u kterých známe zařazení do správné třídy nebo hodnotu jeho cílové proměnné.

Ve své základní podobě je trénování založeno pouze na uložení všech trénovacích dat a výsledek klasifikace je rozhodnut postupem, kdy jsou zjištěny třídy všech k nejbližších sousedů, jak můžeme vidět na obrázku 3.1. Do třídy, která je v tomto výběru nejpočetněji zastoupena je následně zařazen i klasifikovaný prvek, respektive v případě regrese je mu přiřazen průměr hodnot z vybraných sousedů.



Obrázek 3.1: Znázornění algoritmu *kNN* pro *k=1*, *k=2* a *k=3* Zdroj:[1]

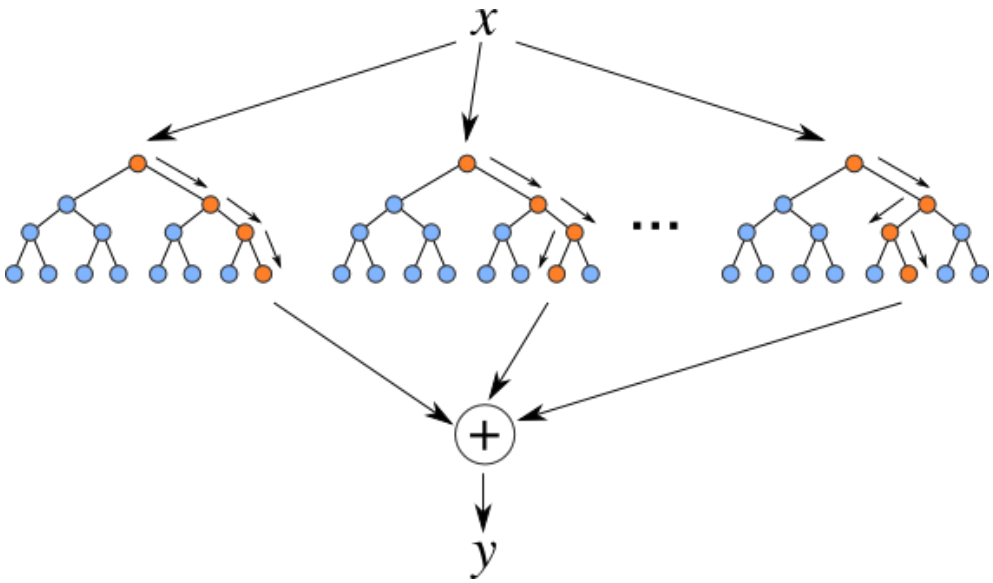
Výběr nejbližších sousedů je vyhodnocen pomocí vzdálenosti mezi jednotlivými vzorky a predikovaným prvkem. Jako základní metrika pro výpočet vzdáleností mezi těmito body se nejčastěji využívá euklidovská vzdálenost:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Jedním z problémů dále spojeným s tímto algoritmem bývá správná volba počtu k nejbližších sousedů. Pro volbu k je možno použít různé heuristiky, obecně však platí, že větší hodnoty k redukují šum při klasifikaci, zato je ale hranice mezi jednotlivými třídami méně zřetelná. V případě rozhodování na základě pouze jednoho nejbližšího souseda *(k=1)* je třída takového souseda převzata jako výsledek predikce, pro tento postup se používá pouze označení *nearest neighbour* [11].

3.2.2 Náhodný les - RF

Algoritmus *náhodný les* taktéž patří k základním algoritmům strojového učení a vychází z rozhodovacích stromů, kdy každý uzel stromu představuje rozhodování podle jedné (vybrané) vlastnosti objektu. Z každého uzlu následně vychází konečný počet hran, které dále rozhodují o zařazení prvku do konkrétní třídy. Počet takovýchto rozhodování je ovlivněn hloubkou jednotlivých stromů a každé cestě stromem od kořene k listu pak odpovídá jedno pravidlo. Pro kořenové uzly je vybrána ta vlastnost prvku, podle které je dané prvky možné maximálně odlišit. Se zvětšující se hloubkou stromu se tato vlastnost snižuje.

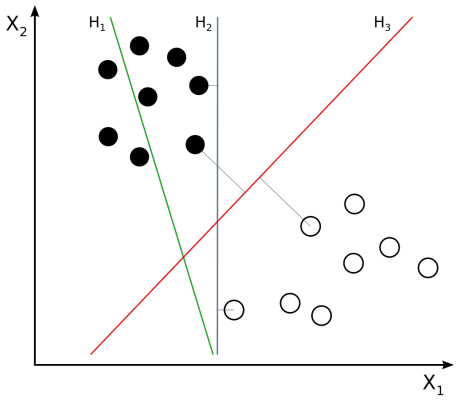


Obrázek 3.2: Znázornění metody *náhodný les* pro 3 stromy s hloubkou 3. Zdroj: [19]

Výhodou oproti klasickým rozhodovacím stromům je vytvoření kolekce stromů (lesa) namísto jednoho stromu. Tento les následně rozhoduje hlasováním o přidělení klasifikovaného prvku do třídy, respektive výpočtem průměrů cílové proměnné z odhadů jednotlivých stromů, pomocí závěrečné sdružovací funkce. Základními parametry pro náhodný les jsou celkový počet a hloubka vytvořených stromů.

3.2.3 Metoda podpůrných vektorů - SVM

Třetí ze zmíněných algoritmů je metoda podpůrných vektorů *(Support Vector Machines)*. Tento algoritmus je alternativní metodou strojového učení a spadá do kategorie tzv. *jádrových algoritmů*. Základní princip je založen na konceptu rozhodujících rovin, který spočívá v hledání nadroviny, která původní prostor příznaků optimálně rozděluje tak, že trénovací data náležící odlišným třídám leží v opačných poloprostorech, a zároveň maximalizuje prostor mezi danou polorovinou a podpůrnými vektory. V případě lineární transformace je tato nadrovina v dvourozměrném prostoru reprezentovaná jako přímka, v třírozměrném jako rovina.



Obrázek 3.3: Znázornění tří možných nadrovin v prostoru, přičemž pouze H_3 je vhodná, neboť maximalizuje prostor mezi prvky a podpůrnými vektory. Zdroj: [19]

Současně ale existují úlohy, které nelze jednoduše řešit lineární transformací. V případě lineárně neoddělitelných tříd jsou na trénovací data aplikovány transformace pomocí jádrových funkcí, které prostor příznaků převádí do prostoru transformovaných příznaků typicky vyšší dimenze. Mezi nejčastější jádrové funkce patří *RBF (Radial Base Functions)*, která je definovaná jako:

$$K(X, X') = \exp(\gamma||X - X'||^2) \tag{3.1}$$

Klasifikátor založený na tomto algoritmu můžeme specifikovat jednak podle použité jádrové funkce, taktéž i podle parametru C *(Penalty factor)*, někdy označovaného jako regularizační konstanta, která definuje přípustnou vzdálenost podpůrných vektorů a vytváří měkké hranice klasifikátoru s tolerancí mírné chyby.

3.3 Vyhodnocení úspěšnosti klasifikace

Informaci o tom, jak úspěšný je trénovaný klasifikační model, můžeme v případě strojového učení s učitelem zjistit pomocí různých metrik. Standardní princip je rozdělení vstupní datové sady na dvě množiny, které jsou vzájemně disjunktní. První množina se použije k natrénování klasifikátoru, druhá testovací množina pak následně k vyhodnocení klasifikátoru.

Ve výsledku nás zajímá, jak si klasifikátor vede na datech, která nezná. Samotné vyhodnocení úspěšnosti klasifikátoru provádíme tak, že klasifikátorem predikované hodnoty porovnáme s předem získanými hodnotami, které považujeme za správné. Pro názornost jsou následující metriky uváděny pro binární klasifikaci. Informace jsou čerpány z [15] a [2].

3.3.1 Kontingenční tabulka

Jednou z nejznámějších metrik pro vyhodnocení klasifikace je kontingenční tabulka[1]. Při binární klasifikaci je možno vzorky datové sady rozdělit do taulky o čtyřech polích, kdy jedna osa popisuje predikované a druhá skutečné hodnoty. Ve výsledku máme dvě pole TP a TN v hlavní diagonále matice popisující správně klasifikované vzorky *(True Positive a True*

[1] kontingenční tabulka je někdy taktéž označovaná jako konfúzní matice či matice záměn

Negative) a zvlášť pole, *FP* a *FN*, pro chybně klasifikované, kdy negativní hodnota byla klasifikována jako pozitivní a naopak *(False Positive a False Negative)*.

		Skutečná hodnota	
		Pozitivní	Negativní
Predikovaná hodnota	Pozitivní	*tp*	*fp*
	Negativní	*fn*	*tn*

Tabulka 3.1: Kontingenční tabulka

3.3.2 Klasifikační metriky

Následné hodnoty vycházejí z kontingenční tabulky 3.1.

- **Správnost** vyjadřuje procento úspěšně predikovaných hodnot vůči všem vzorkům v testovací množině. Rovnici pro výpočet můžeme vyjádřit vzorcem:

$$správnost = \frac{tp + tn}{fp + fn} \tag{3.2}$$

- **Přesnost** udává, jaký je podíl správně predikovaných pozitivních hodnot vzhledem ke všem pozitivním hodnotám v testovací množině a je vyjádřena vzorcem:

$$přesnost = \frac{tp}{tp + fp} \tag{3.3}$$

- **Úplnost** vysvětluje, jaký je poměr správně predikovaných pozitivních hodnot vůči všem pozitivním hodnotám v testovací množině a vychází ze vzorce:

$$úplnost = \frac{tp}{tp + fn} \tag{3.4}$$

- **Míru F1** můžeme interpretovat jako vážený průměr přesnosti a úplnosti. Klasifikátor dosahuje nejlepších výsledků při skóre F1 rovné *1* a nejhorších při *0*. Vzorec pro výpočet je dán:

$$F1 = 2 * \frac{přesnost * úplnost}{přesnost + úplnost} \tag{3.5}$$

3.4 Vyhodnocení úspěšnosti regrese

V případě regrese je situace složitější, neboť nelze o predikovaných hodnotách říci, zdali jsou pravdivé či nepravdivé, jak tomu je v případě klasifikace. Je třeba se ptát na otázku, jak přesné jsou predikované hodnoty vzhledem k správným hodnotám. Dané metriky v následujících dvou sekcích byly čerpány ze zdroje [7].

3.4.1 Střední absolutní chyba a střední kvadratická chyba

Pomocí střední absolutní chyby *(Mean Absolute Error)* a střední kvadratické chyby *(Root mean squared error)* můžeme určit, jak moc se predikované hodnoty liší od skutečných hodnot.

- **Střední absolutní chybu** *(MAE)* vypočítáme podle vzorce:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.6}$$

- **Střední kvadratickou chybu** *(RMSE)* vypočítáme podle vzorce:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.7}$$

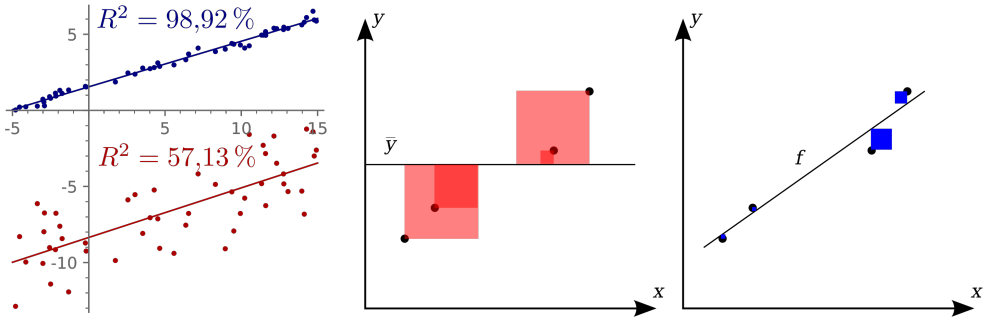
kde n je počet vzorků, y_i značí predikovanou hodnotu a $\hat{y}_i$ skutečnou hodnotu vzorku.

3.4.2 Koeficient determinace

Další možností může být koeficient determinace[2] *(Coefficient of determination)*, který značíme jako R^2. Tento koeficient poskytuje informaci, jak dobře regresní model dokáže vyhodnotit daný rys v procentech. Běžně se počítá na základě podílu sum čtverců rozdílu chyb mezi predikcemi modelu a skutečnými hodnotami SS_{res} a sum čtverců rozdílu chyb predikcí od průměrné hodnoty napříč vzorky SS_{tot}.

Koeficient R^2 je počítán podle vzorce:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.8}$$



Obrázek 3.4: Příklad R^2 a grafické znázornění SS_{res} *(červeně)* a SS_{tot} *(modře)* čtverců [23]

[2] koeficient determinance je taktéž překládaný jako hodnota spolehlivosti či „R kvadrát“

Kapitola 4

Problémy datové sady

V této kapitole si postupně rozebereme konkrétní problémy spojené s užitou datovou sadou a možná řešení.

4.1 Problém nevyvážených dat

Během klasifikace se obecně můžeme setkat s jevem, kdy jsou třídy v dané datové sadě nerovnoměrně zastoupeny. Může se stát, že z celkového množství např. *1000* vzorků, pouze *20* připadá k jedné třídě a zbytek *980* druhé třídě. V tomto případě můžeme říci, že třídy jsou v datové sadě v poměru *2 : 98* a při binární klasifikaci je zastoupení pozitivních hodnot v datové sadě *2%*. Tato nevyváženost se může promítnout ke zkresleným výsledkům klasifikace. Následující dvě části popisují základní techniky, které se v takovém případě využívají, informace jsou čerpány z [21].

4.1.1 Náhodné podvzorkování

Náhodné podvzorkování *(Random Under-Sampling)* je metoda, která spočívá ve vytvoření nové vyvážené datové sady náhodným odebráním vzorků majoritní třídy. V tomto případě snižujeme objem trénovacích dat a tím urychlujeme proces trénování klasifikátoru, na druhou stranu je nutné zmínit, že tímto způsobem můžeme potenciálně přicházet o podstatné informace.

4.1.2 Náhodné převzorkování

Druhou metodou je náhodné převzorkování *(Random Over-Sampling)*, která je komplementární k první zmíněné metodě. Během této metody však nedochází k odstranění vzorků majoritní třídy, ale k náhodné replikaci vzorků minoritní třídy. Ve výsledku tak nedochází ke ztrátě dat, a často daná metoda překonává metodu náhodného podvzorkování, současně ale zvyšujeme pravděpodobnost, že se systém přeučí *(tzv. overfittingu)*, což může mít negativní vliv při testování klasifikátoru na nových datech.

4.2 Selekce rysů

Každý jednotlivý rys zkoumaného vzorku může poskytovat různou vypovídající hodnotu o klasifikované třídě. V případě, kdy vybereme zvlášť všechny vzorky jedné třídy a stejně tak všechny vzorky druhé třídy, můžeme za pomocí různých metod zjistit, jak moc se jednotlivé

rysy liší mezi dvěma třídami a jakou mají jednotlivé rysy vypovídající schopnost o dané třídě. Informace o t-testu jsou čerpány ze zdroje [2].

4.2.1 Studentův T-test

Mezi základní metody vyhodnocení podobnosti, které se využívají ve statistice mezi dvěma vstupními vektory, je metoda *t-test*, též známá jako *Studentův test* [2]. Tato metoda vychází z normálního rozložení a stanovuje míru podobnosti dvou vektorů na základě tohoto rozložení. V práci je implementovaný příklad pro náhodný výběr, který je tvořen dvojicí hodnot; v takovém případě se jedná o *párový t-test*.

V praxi se párový t-test používá jako důkaz hypotézy, zda se předdefinované hodnoty jedné třídy významně liší od hodnot druhé třídy. Prahová hodnota pro důkaz této hypotézy je obecně stanovena na hodnotu *0,05*. Pokud je výsledek t-testu, který nazýváme jako *p-hodnota*, pod touto prahovou hodnotou, daný vzorek se nachází na okrajích normálního rozložení a hypotéza je brána jako pravdivá.

Metoda t-test může být přínosná během klasifikace, neboť je podle ní možno informovaně redukovat mnoharozměrnost trénovacích dat. Umožňuje stanovit, které rysy mohou poskytovat relevantní informace pro predikované cíle a které rysy je přípustné během strojového učení zanedbat.

4.3 Problém kompozičních dat

Obecně pro klasifikační algoritmy není žádoucí, pokud mezi jednotlivými rysy existují vzájemné závislosti. Metagenomická data, na kterých staví současné klasifikátory a klasifikátory implementované v této práci, bohužel tyto závislosti obsahují.

Jak jsme si popsali v sekci 2.6, metagenomická data jsou tvořena různým zastoupením jednotek *OTU* na různých taxonomických úrovních. Jednotlivé taxony jsou tak mezi sebou kompozičně závislé, neboť informace pro každý *taxon* je dána určitým počtem jednotek *OTU*, který je vždy relativní k celkovému množství jednotek *OTU* ve vzorku. Ve výsledku tak každý taxon vždy popisuje informaci, která je silně závislá na celkovém množství jednotek *OTU* ve vzorku. Vzniká tak negativní efekt, kdy větší počet jednotek *OTU* přiřazených jednomu taxonu automaticky implikuje snížení zastoupení ostatních pozorovaných taxonů, což není žádoucí. Články [6] a [22] uvádějí jako vhodné řešení pro tento typ dat aplikování logaritmických transformací.

4.3.1 Normalizace

Před započetím samotné logaritmické transformace je vhodné data normalizovat, neboť rozdílné množství celkového počtu OTU napříč vzorky by mohlo mít negativní vliv na následnou klasifikaci. Výsledek normalizace pak nabývá hodnot z intervalu $\langle 0,\ 1\rangle$. Mezi základní přístupy normalizace patří *min-max normalizace*, která je dána vzorcem:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

kde *x=(x1,...,xn)* je původní vektor a z_i normalizovaný vektor [15].

4.3.2 Centred Log-Ratio - CLR

Doporučovanou logaritmickou transformaci pro tento typ dat je dle článku [6] logaritmická transformace CLR *(Centered Log-Ratio)*. Tuto logaritmickou transformaci můžeme vypočítat za pomoci vzorce:

$$clr(x) = \ln\left[\frac{x_1}{g_m(x)}, \ldots, \frac{x_D}{g_m(x)}\right] \tag{4.2}$$

kde

$$g_m(x) = (\prod_{i=1}^{D} x_i)^{1/D} \tag{4.3}$$

je geometrický průměr vektoru x [17].

4.4 Problém vizualizace mnoharozměrných dat

Běžně každý rys reprezentuje jednu dimenzi v grafu. V ideálním případě, tedy na základě pouze dvou nebo tří informací o daném vzorku, je možné dané prvky jednoduše promítnout do dvourozměrného či trojrozměrného prostoru o osách x, y a z, které by dané rysy reprezentovaly.

Komplikace nastává v případě, kdy počet rysů převyšuje tento počet dimenzí, což v reálném trojrozměrném světě představuje problém. Tuto situaci můžeme kompenzovat promítnutím postupně více grafů s osami podle jednotlivých rysů, kde využijeme celou informaci daného vzorku, celková vypovídající hodnota by ale mohla být těžko interpretovatelná.

V následujících dvou sekcích si představíme metody lineární transformace, které se daným problémem zabývají, informace jsou čerpány ze zdrojů [14], [18] a [15].

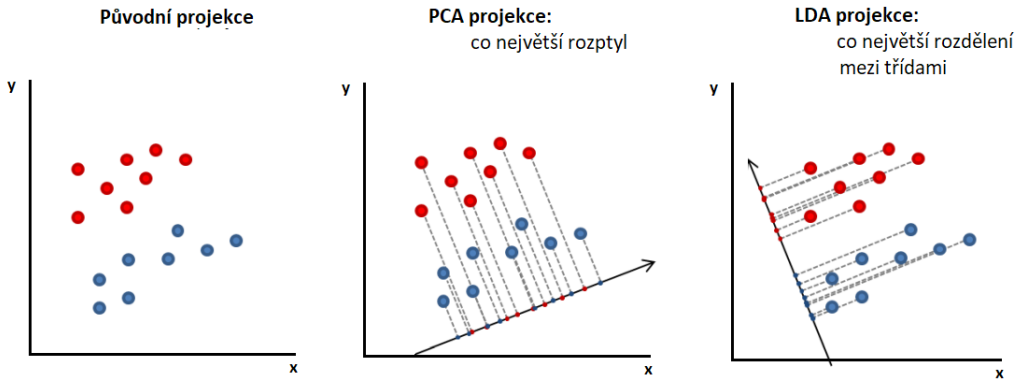
4.4.1 Analýza hlavních komponent – PCA

Analýza hlavních komponent *(Principal Component Analysis)* je neinformovaná metoda, která se snaží redukovat počet dimenzí v euklidovském prostoru tak, aby došlo k minimální ztrátě informace. Tato metoda je založena na transformaci původních znaků $x_1, x_2, ...x_n$ na nové, nekorelované proměnné $y_1, y_2, ...y_n$, které nazýváme *hlavní komponenty (Principal Components)*, kde komponenta představuje lineární kombinaci původních znaků.

Hlavní komponenty jsou seřazeny dle důležitosti, tj. dle klesajícího rozptylu, od největšího k nejmenšímu. Každá z komponent je tak ohodnocena skórem, které je stanoveno podílem proměnlivosti a vypovídá o tom, kolik procent celkové informace je charakterizováno danou komponentou. Součet všech hodnot podílů proměnlivosti je roven jedné. Standardním využitím je redukce počtu znaků bez velké ztráty informace s využitím prvních dvou nebo tří komponent, které se dále využívají jako osy k projekci do roviny respektive prostoru.

4.4.2 Lineární diskriminační analýza – LDA

Lineární diskriminační analýza taktéž redukuje počet dimenzí. Na rozdíl od PCA je LDA informovaná metoda, která hledá určité diskriminátory v podprostoru, které mají za úkol co nejefektivněji oddělit dané třídy namísto rozptylu pro každý vzorek zvlášť, jak lze vidět



Obrázek 4.1: Princip projekce dat zvlášť metodou PCA a LDA. Zdroj: [18]

na obrázku 4.1. Pomocí diskriminační analýzy dostaneme odpověď na otázku, do jaké míry se námi předdefinované třídy liší ve znacích, které máme k dispozici.

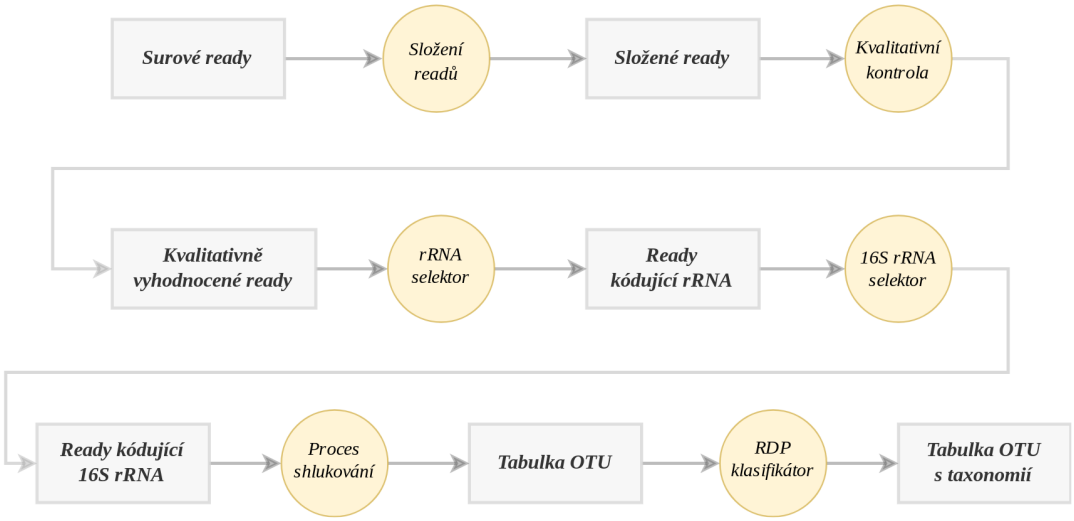
Princip LDA spočívá v redukování jednotlivých dimenzí za pomocí diskriminační funkce, která od sebe co nejlépe dané třídy separuje. Cílem je identifikovat takové nezávislé proměnné, které nejlépe odlišují jednotlivé třídy objektů. Tyto nezávislé proměnné nazýváme diskriminátory. V případě binární klasifikace se tak pomocí diskriminační funkce převede dvourozměrný problém zařazování na problém jednorozměrný s ohledem na dané třídy.

Kapitola 5

Tvorba datových sad

Prvotním krokem práce bylo získat vhodná referenční data, která by bylo možné využít k natrénování klasifikátoru. Množství použitelných datových sad je omezené, neboť nezbytnou součástí těchto dat musí být i rozšiřující informace o každém vzorku, na základě kterých by bylo možné provádět další klasifikaci, příp. regresi zvolených životních atributů.

Jako jedny z mála vhodných se ukazují záznamy projektu *American Gut*[1], které jsou k dispozici v databázích Evropského bioinformatického institutu *EBI*[2]. Ke každému z těchto vzorků je dostupný vstupní dotazník[3], který obsahuje informace o stravovacích návycích, alergiích, věku, váze, bmi aj. Dostupné datové soubory jsou demultiplexovaná čtení[4], která jsou rozdělena zvlášť do jednotlivých souborů a dále zpracována.



Obrázek 5.1: Proces zpracování surových readů na serverech EBI

[1] veřejný projekt vzbuzující iniciativu pro zmapování lidského mikrobiomu založený Robem Knightem
[2] data dostupná na https://www.ebi.ac.uk/metagenomics/studies/ERP012803
[3] příklad dotazníku je možné dohledat na https://www.ebi.ac.uk/ena/data/view/SAMEA3607595
[4] soubory obsahující vždy jeden vzorek, běžně je jinak sekvenováno více vzorků současně

Zpracování surových metagenomických dat zahrnuje operace náročné na výpočet a zpracování probíhá na serverech *EBI*. Jednotlivé kroky procesu zpracování jsou znázorněny v diagramu 5.1. Výsledkem jsou tabulky *OTU* ve formátu *BIOM*, popisující shluky sekvencí se standardní *97%* podobností, kterým je přiřazené taxonomické zastoupení podle referenční databáze *Greengenes 13_8*[5].

5.1 Zpracování datových sad

Celkem bylo hromadně staženo zhruba *6500* souborů *BIOM* za pomoci nástroje *ebi-metagenomics*[6]. Pro další zpracování těchto souborů a současně získání informací z dotazníků byly využity skripty nacházející se na přiloženém datovém médiu ve složce `data/scripts/`.

Atributy byly po jednom staženy ve formátu *XML*, ze kterých byly dále extrahovány relevantní informace a společně uloženy do jednoho *CSV* souboru. Taxonomické zastoupení na rozdílných úrovních od *L2* po *L6* bylo z každého ze souborů *BIOM* extrahováno s využitím nástroje *QIIME* ve verzi *1.9*. Toto taxonomické vyhodnocení bylo následně sdruženo do pěti souborů *CSV* odpovídajících každé taxonomické úrovni zvlášť.

Ve výsledku tak jsou datové sady tvořeny šesti soubory. Jedním souborem obsahujícím informace z dotazníku pro každý vzorek, který je společný pro všechny taxonomické data, k tomu dále pěti soubory reprezentovaných jako tabulky *OTU*. Tato data byla následně podstoupena další analýze.

5.2 Základní filtrace datových sad

Každý vzorek datové sady si můžeme představit jako vektor celých čísel, který vždy v součtu dává celkové množství jednotek *OTU* ve vzorku. Množství těchto jednotek se u jednotlivých vzorků liší. Tento jev je způsoben rozdílnou hloubkou sekvenování, díky které je získán rozdílný počet řetězových sekvencí.

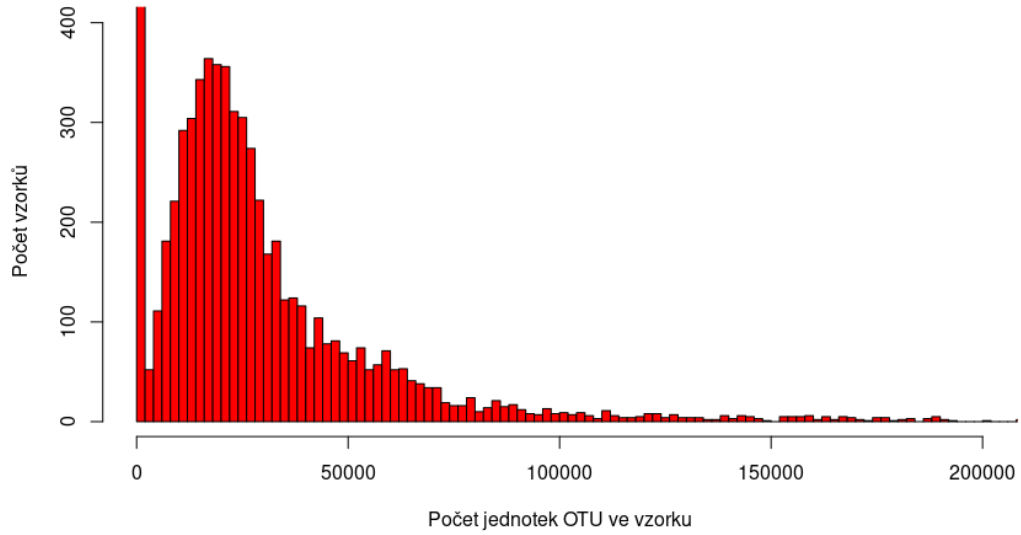
Nižší počet sekvencí ve vzorku tak dává za vznik i nižšímu počtu jednotek *OTU* a naopak. V případě nižšího počtu původních sekvencí tak nemusely být dostatečně rozpoznány všechny přítomné bakteriální taxony [6], v opačném případě, kdy původní vzorek obsahuje znatelně více sekvencí, taktéž nabývá i větší různorodosti jednotlivých taxonů. Rozdílné celkové množství *OTU* ve vzorcích původní datové sady popisuje histogram 5.2.

Zde je možné si všimnout poměrně velkého množství chybně zpracovaných tabulek *OTU*, kdy je množství těchto jednotek ve vzorku velmi malé a blíží se nule. Stejně tak jsou v základní datové sadě zastoupeny vzorky obsahující počet jednotek *OTU* v řádu statisíců (až *500 000*).

Z těchto vzorků byly dále vybrány pouze vzorky, obsahující množství jednotek *OTU* od *5000* po *40000*, které jako celek nejvíce odpovídají normálnímu rozložení. Ostatní vzorky byly z datové sady odstraněny.

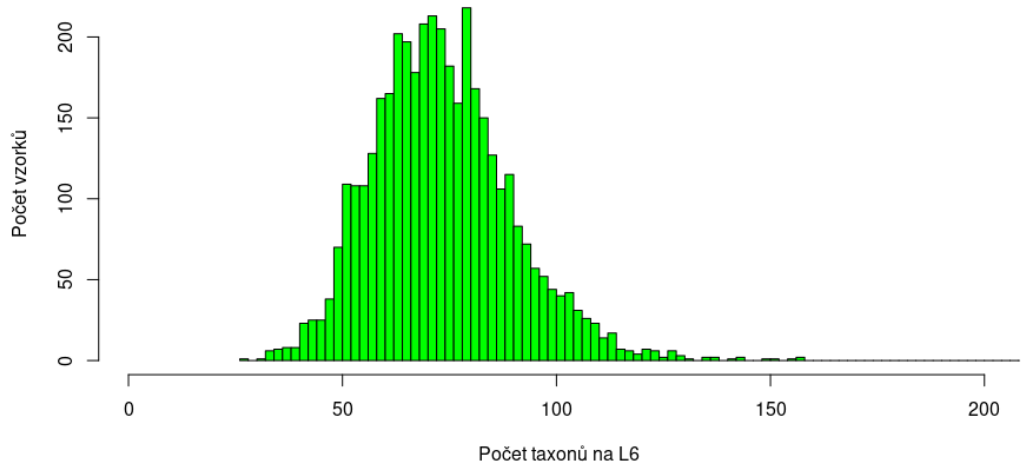
[5] databáze Geengenes je dostupná na https://greengenes.secondgenome.com/

[6] konzolová aplikace dostupná na https://github.com/ProteinsWebTeam/ebi-metagenomics



Obrázek 5.2: Histogram popisující množství jednotek OTU ve vzorcích

Běžně se stejně tak z tabulek OTU odstraňují i ty taxony, jejichž existence ve vzorku je podmíněna pouze jednou či dvěma jednotkami OTU, neboť u těchto taxonů nelze dokázat jejích opravdovou přítomnost ve vzorku a ve většině případech se jedná jen o chybu sekvenace. Takovéto taxony označujeme v tabulce OTU jako *singletony* či *doubletony*. Datová sada byla dále zkontrolována i z hlediska množství jednotlivých taxonů. Histogram 5.3 ukazuje jednotlivé množství přiřazených taxonů pro každý vzorek zvlášť na úrovni *L6*.



Obrázek 5.3: Histogram popisující množství taxonů přítomných ve vzorcích

5.3 Zvolení atributů

Jako atributy vhodné ke klasifikaci se ukazují především dotazníkem definované stravovací návyky pacienta *(masožravec, vegetarián)* [4], případně dále alergie na lepek či laktózu [8], věk [16] a body mass index [25], jako poslední klasifikační atribut bylo zvoleno pohlaví jedince [25]. V případě klasifikace tyto atributy rozdělují datovou sadu na dvě třídy:

- `diet_type` - rozdělení podle různých stravovacích návyků na *masožravce* a *vegetariány*
- `sex` - rozdělení podle pohlaví na *muže* a *ženy*
- `gluten` - rozdělení *ano, ne* podle alergie na lepek
- `lactose` - rozdělení *ano, ne* podle alergie na laktózu

Atributy pro regresní analýzu můžou být buď celočíselné, nebo spojité hodnoty jako například:

- `age_years` - *věk* pacienta
- `bmi` - hodnota *body mass indexu* pacienta

Kapitola 6

Návrh klasifikátoru

V této kapitole si popíšeme základní návrh prediktoru před započetím implementace. Následná implementace se snaží zachovat zde zmíněné principy, je však uzpůsobena podle analýzy výsledků prediktoru (viz 7 a 8). Nástroj by měl být dle zadání práce schopen pracovat s několika typy klasfikátorů. Z čerpané teorie se jako vhodné nabízí postupně metody *k nejbližších sousedů (kNN)*, *náhodný les (RF)* a *metoda podpůrných vektorů (SVM)*. Pomocí klasifikátoru by mělo být možné vyhodnotit, které taxonomické úrovně jsou pro zvolený atribut nejvlivnější.

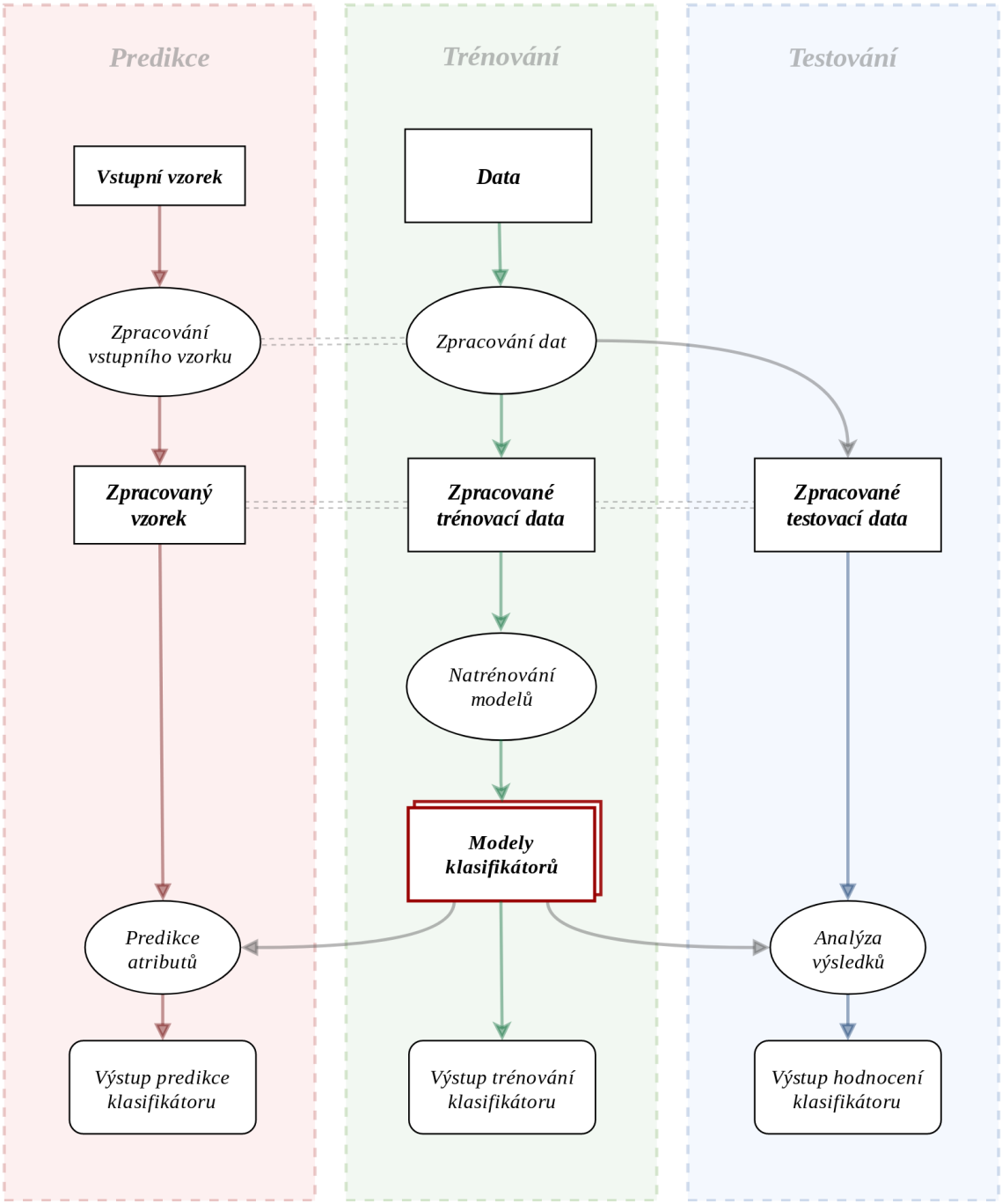
6.1 Rozdělení nástroje

Z pohledu klasifikace by bylo vhodné nástroj rozdělit do několika fází, které na sebe logicky navazují. Klasifikátor by tak mohl poskytovat ucelené informace jak o **trénování**, tak **testování** a finální **predikci**. Zvlášť jsou pak popsány části pro načtení vstupních dat, předzpracování dat dle parametrů, natrénování modelů, testování klasifikátoru a výstup predikce.

Vstupní data se liší podle aktuálního stavu nástroje:

1. **Natrénování klasifikátoru** pracuje se základní datovou sadou, vstupem od uživatele je množina parametrů potřebných k zpracování dat a natrénování klasifikátoru.
2. **Při testování klasifikátoru** je vstupem složka obsahující jednotlivé natrénované modely, které je možné použít pro testování a následně pro predikci.
3. **Před započetím predikce** je vstupem jeden vzorek dat reprezentovaný jako tabulka *OTU* ve formátu *BIOM* s přiřazeným taxonomickým zastoupením.

Základní zpracování dat spočívá v odstranění těch vzorků, u kterých je celkové množství jednotek *OTU* nižší nebo případně vyšší, než nastavený práh. Dále jsou data filtrována podle definovaných tříd z hodnot dotazníku, kdy nevalidní hodnoty atributů jsou z datové sady před trénováním konkrétní třídy odstraněny.



Obrázek 6.1: Blokové schéma návrhu klasifikátoru s rozděleném do tří částí

Předpokladem je možnost natrénování dat pomocí různých klasifikátorů založených na algoritmech strojového učení rozebraných v sekci 3.2. Každý algoritmus je možno specifikovat vstupními parametry:

- Pro algoritmus *kNN* se jedná o:
  - počet nejbližších sousedů n
- dále algoritmus *RF* podle:
  - počtu rozhodovacích stromů t
  - hloubky d
- algoritmus *SVM* pomocí:
  - jádrové funkce k
  - faktoru penalizace c

Dané vzorky by mělo být taktéž možno klasifikovat na různé taxonomické úrovní od *L2* po *L6*, pro každý zvolený atribut zvlášť. V případě delší doby trénování jednotlivých modelů by bylo vhodné natrénované modely opakovaně využít.

Vyhodnocení klasifikátoru by mělo být možné realizovat zvlášť na testovací sadě. Pro každý z natrénovaných modelů s různým nastavením parametrů je výstupem testování soubor metrik, pro klasifikaci *kontingenční tabulka*, *správnost*, *přesnost*, *úplnost* a *míra F1*, které jsou rozebrány v sekci 3.3, pro regresi pak *střední absolutní chyba*, *střední kvadratická chyba* a *koeficient determinance* ze sekce 3.4.

Predikce by měla vycházet z natrénovaných modelů, které byly vyhodnoceny jako nejpřesnější. Podle získaných poznatků se může jednat o modely založené na datových sadách různé taxonomické úrovně, různých typů klasifikátorů s různými parametry.

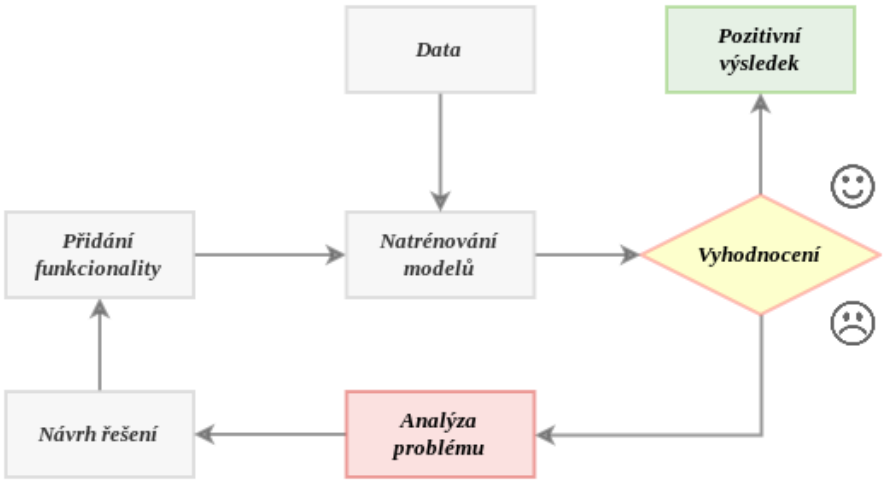
Výstup nástroje je rozdělen do tří částí a zvlášť popisuje trénovaní modelů, kdy je současně výpisem veškerý postup zpracování datové sady, testování přesnosti modelů s výpisem jednotlivých metrik a hodnoty predikce.

Kapitola 7

Implementace

V této kapitole je popsána implementace nástroje navrženého v předchozí kapitole. Pro implementaci byl zvolen jazyk Python ve verzi *3.6* s využitím knihoven *Numpy* a *Pandas* pro práci s vektorovými daty a *Sci-kit learn* poskytující algoritmy strojového učení. Další využité prostředky jsou rozebrané v sekci 7.7.

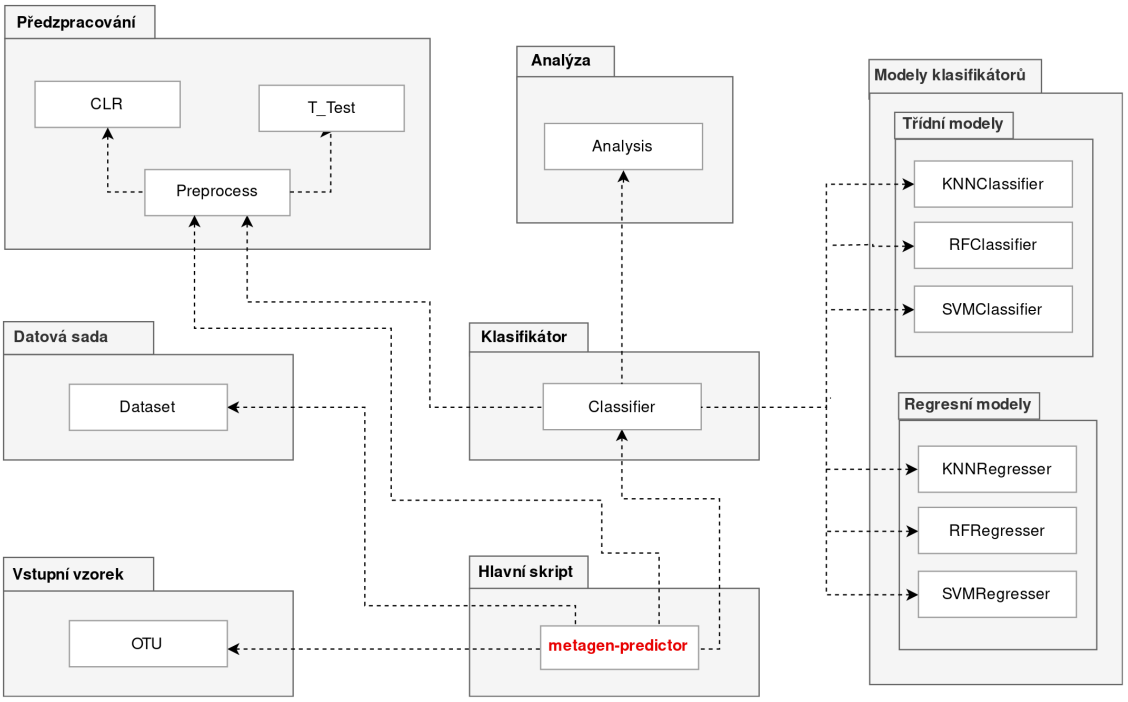
Na začátku byla implementována základní funkcionalita nástroje, tedy rozdělení nástroje do tří částí zvlášť pro *trénování*, *testování* a *predikci*, získání vstupu, zpracování dat, natrénování modelů klasifikátorů a realizace výstupu. Další proces implementace úzce souvisí s analýzou výsledků a je reakcí na poznatky získané při testování natrénovaných modelů v kapitole 8. Nejrozsáhlejším vývojem prošla část pro zpracování vstupní datové sady, která byla průběžně doplňována o nové techniky popsané v kapitole 3 a 4. S rozšiřující částí předzpracování dat přibývalo i množství informací poskytovaných na výstupu klasifikátoru. Proces pozdější implementace lze znázornit diagramem 7.1.



Obrázek 7.1: Diagram: Proces rozšiřování implementace nástroje

7.1 Implementované části nástroje

Z pohledu programátora je nástroj rozdělen do více vzájemně komunikujících částí, které spojuje hlavní soubor `metagen-predictor.py`. Tento soubor reprezentuje vstupní část nástroje, které je možno zadat parametry a soubory pro trénování modelů, testování i predikci.



Obrázek 7.2: Schéma finální implementace nástroje

7.2 Proces načtení dat do prediktoru

Nástroj pracuje s třemi druhy vstupních souborů:

- zdrojové datové sady ve formátu *CSV*
- soubory natrénovaných modelů ve formátu *JOBLIB*
- vstupní vzorek ve formátu *BIOM* verze *2.0* a vyšší, obsahující jeden vzorek, který má přiřazené taxonomické zastoupení podle referenční databáze *Greengenes 13_8*

Vstupní datové soubory a vzorek určený k predikci jsou po dobu běhu programu reprezentovány jako instance třídy *Dataset*, se kterou pracuje převážná část implementovaných metod.

7.3 Proces zpracování vstupních dat

Zpracování vstupních dat se skládá jak ze základní filtrace vstupní datové sady popsané v návrhu (6.1), tak z aplikování přídavných metod pro normalizaci, transformaci a selekci rysů. Celkově se tak proces filtrace a zpracování vstupních dat skládá z několika kroků:

1. Filtrace vzorků uživatelem specifikovaného intervalu celkového počtu jednotek *OTU* ve vzorku
2. Kontrola atributů a sjednocení vzorků do klasifikačních tříd
3. Odebrání taxonů s celkovým výskytem v datové sadě nižším než specifikovaná procentuální hodnota, příp. zachování pouze n taxonů
4. *Normalizace* vzorků (viz sekce 4.3.1)
5. Logaritmická *transformace* vzorků pomocí *CLR* (4.3.2)
6. Odebrání vzorků, které nemají přesně definovanou hodnotu zvoleného atributu
7. Rozdělení datové sady podle tříd a náhodné *podvzorkování* či *převzorkování* (4.1)
8. Selekce rysů s největší vypovídající hodnotou o dané třídě pomocí párového *t-testu* (4.2)

Kroky 1 až 5 jsou zpracovány nezávisle na modelu třídního či regresního klasifikátoru. Kroky 6 až 8 jsou specifické pro třídní klasifikátory a aplikované těsně před natrénováním modelu podle zvolených atributů určených k predikci. Předzpracování datové sady je řešeno třídami *Preprocess*, *CLR* a *T_Test*.

7.4 Proces natrénování modelů

Natrénované modely jsou reprezentovány jako instance jednotlivých tříd, jak lze vyčíst v diagramu 7.2 a jsou charakteristické metodami *train()* a *predict()*.

Implementovaný nástroj trénuje tyto modely zvlášť nejen pro každý algoritmus strojového učení, ale i pro každý zvolený atribut určený k predikci. Výsledné modely lze stejně tak rozdělit i podle taxonomické úrovně použité datové sady. Logika tvorby jednotlivých modelů je popsána v třídě *Classifier*. V reálné situaci vzniká poměrně velké množství natrénovaných modelů, které je vhodné pro budoucí účely zachovat.

Tento problém je řešen ukládáním instancí těchto tříd do souborů za pomoci knihovny *JOBLIB*. Cesta k těmto souborům je dána parametrem `--model-path` hlavního skriptu.

7.5 Způsoby vyhodnocení

Návrh v kapitole 6.1 počítá s vyhodnocením zvlášť nad testovacími daty z důvodu možnosti oddělit výstup testování od trénování. Základní přístup implementovaného nástroje standardně zachovává popsaný princip náhodným rozdělením datové sady v uživatelem specifikovaném poměru pomocí parametru `--training-split`. Jako rozšíření nástroj taktéž využívá křížovou validaci. V případě zadání parametru `--cross-validation` je možné využít rozdělení celé datové sady na K částí, kdy je proces trénování a testování opakován právě K-krát pro zpřesnění výsledků.

7.6 Analýza datových sad

Během analýzy výsledků vznikla snaha získat taktéž vizuální zpětnou vazbu o zpracovaných datech, jak je popsáno v sekci 4.4. Za tímto účelem byly implementovány postupně dvě třídy. Jako první byla implementovaná třída *PCA*. Později byla vizualizace doplněna o třídu *LDA*. Grafy výsledných analýz během testování jsou uloženy do složky s natrénovanými modely.

7.7 Využité prostředky

Při implementaci byly využity následující prostředky:

- **SciPy**[1]

  Knihovna *SciPy* poskytuje prostředí pro vědecké výpočty. Součástí této knihovny je modul pro práci s vektorovými daty *Numpy*, datovými sadami *Pandas* a modul pro vizualizaci *Matplotlib*.

- **Sci-kit learn**[2]

  Knihovna *sci-kit learn* je nejvyužívanější knihovnou pro strojové učení v Pythonu. Tato knihovna obsahuje širokou škálu algoritmů strojového učení jak pro klasifikaci, tak regresi.

- **Scikit-bio**[3]

  Knihovna *scikit-bio* je základní knihovou pro bioinformatické výpočty v Pythonu, poskytující datové struktury a algoritmy pro vědecké výpočty.

- **biom-format a biom**[4]

  Projekt *biom-format* poskytuje standard pro soubory *BIOM (Biological Observation Matrix)*, které uchovávají informace o tabulkách *OTU*. Součástí projektu je i knihovna pro jazyk Python, ze které bylo čerpáno při načítání vstupního vzorku prediktoru.

- **joblib**[5]

  Knihovna *joblib* umožnuje vytvářet zřetězené operace v jazyce Python. Taktéž je autory *sci-kit learn* doporučována pro ukládání natrénovaných modelů klasifikátorů.

[1] dokumentace dostupná na stránkách https://www.scipy.org/
[2] dokumentace dostupná na stránkách https://scikit-learn.org/stable/
[3] dokumentace dostupná na stránkách http://scikit-bio.org/
[4] dokumentace dostupná na stránkách http://biom-format.org/
[5] dokumentace dostupná na stránkách https://joblib.readthedocs.io/en/latest/

Kapitola 8

Analýza výsledků

V této kapitole jsou rozebrány výsledky klasifikace. První část poukazuje na počáteční výsledky na nezpracovaných datech a nutnost rozšíření implementace. U následujících sekcí jsou již postupně aplikována přídavná předzpracování dat. V zde reprezentovaných výsledcích, byl zvolen atribut *diet_type*, rozdělující datovou sadu do dvou kategorií na *všežravce* a *vegetariány*, který by měl podle odborné literatury nejvíce nasvědčovat možnosti, že data jsou od sebe vzájemně separovatelná.

Jako první se ukazuje se, že výsledky přiřazení jednotlivých vzorků do tříd jsou zkresleny nevyvážeností datové sady. Z toho důvodu bylo přistoupeno k implementaci metod *náhodného podvzorkování* a *náhodného převzorkování* (4.1), pomocí kterých je při každém výsledku testování vždy podvzorkována majoritní třída. Při využití celé datové sady s počtem jednotek *OTU* v rozmezí *5000* až *40000* bylo celkem k dispozici *3976* vorků. Z původních *3450* vzorků masožravců pak byl pak v každém kroku náhodně vybrán počet odpovídající celkovému počtu vegetariánů, jak můžeme vidět v tabulce 8.1.

	Původní počet	Podvzorkovaný počet
Masožravci	3450	418
Vegetariáni	418	418

Tabulka 8.1: Nevyvážená datová sada a podvzorkování třídy masožravců

Vyhodnocení z počátku probíhalo na nejnižší dostupné taxonomické úrovni, kde je poskytována největší informace o každém vzorku. Dalším cílem bylo vyhodnocení úspěšnosti klasifikace na ostatních taxonomických úrovních zpětně od *L5* až po *L2*. Při vyhodnocení modelů poskytujících nejpřesnější výsledky byla stejná konfigurace vyzkoušena i pro jiné životní atributy. V závěru jsou výsledky doplněny o vizualizaci pomocí analýz *PCA* a *LDA*, kdy bylo zkoumáno, jaký vliv na klasifikaci má hloubka sekvenování počátečních řetězcových sekvencí.

Původní verze nástroje byla natrénovaná na základních datech, kde vstupní vzorky měly rozdílné množství jednotek *OTU*. Tento postup se nezdál být příliš efektivní a výsledky na testovací sadě v průměru nedosahovaly přesnosti větší než *53%*. Dalším krokem bylo aplikování normalizace *min-max*, kdy bylo množství jednotek OTU u všech vzorků sjednoceno na hodnotu v intervalu *0* až *1*. Postup trénování a testování celé datové sady se opakoval opět s rovněž nepříliš přesvědčivými výsledky.

Rozšiřujícím krokem byla aplikace jedné z logaritmických transformací nad daty, která je doporučována v případě vzájemně závislých kompozičních dat článkem [6]. Konkrétně byla vybrána transformace *CLR*, která je popsána v sekci 4.3.2. Obecně známý problém logaritmických transformací je, že logaritmické funkce nejsou definované pro nulové hodnoty a limitě se blíží k mínus nekonečnu. Na druhou stranu tabulky *OTU* by se daly popsat jako velmi řídké matice, které jsou charakteristické vysokým množstvím nulových hodnot. V takovém případě je doporučeno nulové hodnoty nahradit nízkými pseudohodnotami, tedy hodnotami, které jsou dostatečně blízké nule pro oddělení těchto hodnot od hodnot nenulových [6].

8.1 Dopad normalizace dat

Jak je možné vyčíst z následujících tabulek, aplikace normalizace může mít na budoucí klasifikaci mírný vliv. Zobrazené výsledky jsou pro úroveň *L6* v rozmezí celkového počtu jednotek *OTU* mezi *5000* a *40000* a s využitím klasifikátoru *kNN* a pěti sousedy.

- **Výsledky klasifikace po CLR transformaci:**

	Správnost	Přesnost	Úplnost	Míra F1
Bez normalizace	0.52	0.52	0.52	0.52
S normalizací	0.54	0.55	0.53	0.54

Tabulka 8.2: Tabulka popisující rozdílné výsledky

8.2 Selekce rysů datové sady

Jako další postup bylo přistoupeno k výběru pouze omezeného počtu rysů užitých k predikci. Nejdříve bylo přistoupeno k neinformovanému redukování počtu dimenzí ořezáním datové sady a zachováním pouze n taxonů, které jsou mezi vzorky nejzastoupenější buď podle přesného počtu rysů, nebo podle procentuálního zastoupení mezi vzorky. Dle dosažených výsledků nelze jednoznačně prokázat, že by selekce rysů pomocí informovaných metod, jako například *t-test*, dosahovala lepších výsledků. Užití těchto metod sice vedlo k mírnému zlepšení výsledné predikce, výsledná přesnost klasifikátoru není ani v takovém případě uspokojivá. Pro názornost je při výsledcích v tabulkách byla použita stejná konfigurace jako v předcházejícím vyhodnocení.

- **Ořezání datové sady**

  Taxonomická úroveň: *L6*

  Taxon minimálně zastoupen v n vzorcích: *5%*

  Počet rysů před a po ořezání: *743 / 129*

	Správnost	Přesnost	Úplnost	Míra F1
Před	0.54	0.54	0.52	0.53
Po	0.57	0.58	0.51	0.54

- **Metoda t-test**

  Taxonomická úroveň: *L6*

  Prahová hodnota: *0,05*

  Počet rysů před a po ořezání: *743 / 44*

	Správnost	Přesnost	Úplnost	Míra F1
Před	0.54	0.54	0.52	0.53
Po	0.56	0.57	0.49	0.53

8.3 Vyhodnocení použitých algoritmů a jejich parametrů

Dalším krokem bylo experimentování s různými parametry klasifikátorů, které by mohlo vést ke zlepšení nepříliš přesvědčivých výsledků. Na použitou datovou sadu byla již aplikována normalizace, transformace *CRL* a selekce rysů pomocí t-testu. Experimenty probíhaly současně na různých taxonomických úrovních s využitím křížové validace. Součástí tabulek je i průměrná přesnost klasifikátoru pro konkrétní taxonomickou úroveň a průměr jednotlivě naměřených odchylek.

- **KNN klasifikátor:**

  U klasifikátoru založeném na algoritmu *kNN* bylo experimentováno s rozdílným množstvím nejbližších sousedů, což by mohlo vést k redukci šumu. Tato skutečnost se ale neprojevila.

	$n = 1$	$n = 3$	$n = 5$	$n = 7$	$n = 9$	$\bar{x}$	$\bar{d}$
$L2$	0.54	0.56	0.53	0.49	0.49	0.49	0.05936
$L3$	0.49	0.48	0.47	0.50	0.48	0.46	0.04264
$L4$	0.50	0.54	0.53	0.52	0.57	0.53	0.07254
$L5$	0.53	0.54	0.55	0.52	0.54	0.54	0.03453
$L6$	0.53	0.58	0.56	0.56	0.56	0.55	0.08356

Tabulka 8.3: Tabulka popsisující *přesnost* klasifikace pro rozdílný počet sousedů n

- **RF klasifikátor:**

  V případě experimentování s parametry u klasifikátoru založeném na algoritmu *RF* bylo experimentováno jak s hloubkou jednotlivých stromů, tak s celkovým počtem stromů. V následující tabulce je hloubka stromů *10*, tedy stejně pro všechny zobrazené výsledky, neboť vliv rozdílné hloubky stromů byl zanedbatelný. Pozornost byla v tomto případě věnována různému počtu stromů.

	$t = 20$	$t = 40$	$t = 60$	$t = 80$	$t = 100$	$\bar{x}$	$\bar{d}$
$L2$	0.48	0.51	0.49	0.52	0.50	0.50	0.06342
$L3$	0.51	0.53	0.52	0.50	0.49	0.51	0.07356
$L4$	0.52	0.50	0.49	0.51	0.51	0.51	0.05317
$L5$	0.53	0.55	0.54	0.56	0.58	0.55	0.09775
$L6$	0.55	0.54	0.53	0.57	0.55	0.55	0.06543

Tabulka 8.4: Tabulka popsisující *přesnost* klasifikace pro rozdílný počet stromů t

- **SVM klasifikátor:**

  V případě klasifikace pomocí metody podpůrných vektorů byl vyzkoušen nejdřív lineární klasifikátor, později i klasifikátor s jádrovou funkcí *rbf (výsledky lineárního klasifikátoru jsou v tabulce znázorněny zeleně, červeně výsledky s využitím jádrové funkce rbf)*. Současně bylo experimentováno s rozdílnou regularizační konstantou C, vytvářející měkké hranice klasifikátoru s tolerancí mírné chyby.

	$c = 1$	$c = 10$	$c = 100$	$c = 1$	$c = 10$	$c = 100$	$\bar{x}$	$\bar{d}$
$L2$	0.53	0.51	0.54	0.55	0.50	0.51	0.52	0.07478
$L3$	0.54	0.52	0.55	0.53	0.51	0.54	0.53	0.03094
$L4$	0.53	0.55	0.52	0.53	0.52	0.57	0.54	0.04116
$L5$	0.53	0.54	0.53	0.58	0.56	0.55	0.56	0.04944
$L6$	0.53	0.54	0.52	0.62	0.58	0.53	0.56	0.07253

Tabulka 8.5: Tabulka popisující *přesnost* klasifikace při rozdílném parametru C

8.4 Vliv různých životních atributů

Jako nejpřínosnější taxonomická úroveň se zdá být úroveň *L6*, případně z části i *L5*. Na úrovni *L6* byla pak vyzkoušena i predikce jiných životních atributů, jako jsou potravinové alergie na lepek či laktózu a pohlaví. Podle předchozích experimentů bylo využito klasifikátoru založeném na metodě podpůrných vektorů s *rbf* jádrovou funkcí a parametrem *C=1*. Současně bylo experimentováno i s kvalitativními hodnotami, jako je hodnota bmi daného jednice a jeho věk, které by taktéž mohly být ovlivněny složením lidského mikrobiomu. Souhrn vyhodnocení těchto atributů popisuje následující tabulka 8.6. Pro klasifikované atributy se výsledky pohybovaly v podobném rozmezí jako u stravovacích návyků v předchozích experimentech. V případě regresní analýzy nástroj nedokázal rozumně predikovat dané hodnoty a výsledky byly nepřiměřeně zkresleny.

- **Vliv různých životních atributů**

	Správnost	Přesnost	Úplnost	Míra F1	MAE	RMSE	R2
diet_type	0.52	0.63	0.63	0.58	-	-	-
gluten	0.47	0.48	0.52	0.46	-	-	-
lactose	0.51	0.50	0.53	0.52	-	-	-
sex	0.58	0.61	0.59	0.58	-	-	-
age_years	-	-	-	-	21.8	25.8	0.05
bmi	-	-	-	-	3.54	5.38	0.01

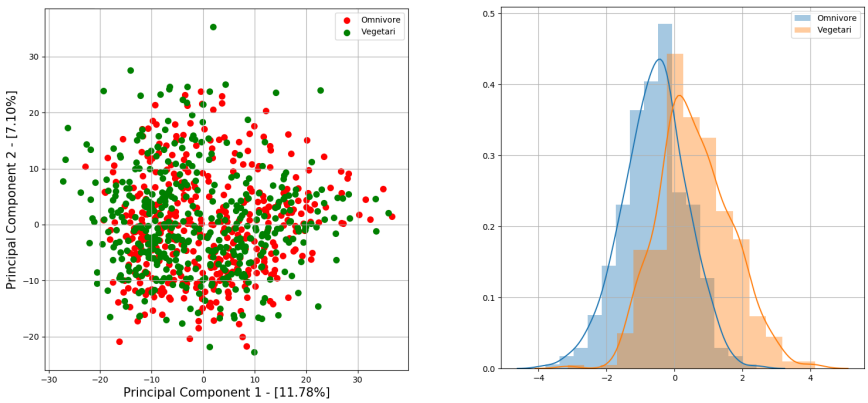
Tabulka 8.6: Tabulka popisující *přesnost* pro různé životní atributy

8.5 Analýza datové sady pomocí PCA a LDA

Další možností byla vizualizace původní datové sady pomocí analýz *PCA* a *LDA*. Výsledné grafy lze nalézt na přiloženém CD ve složce `models/analysis/`. Při klasifikaci na celé datové sadě s množstvím jednotek OTU od 5000 po 40000 se jednotlivé třídy nezdají býti snadno oddělitelné, jak taktéž nasvědčují grafy PCA i LDA analýzy 8.1.

Experimenty na původním rozsahu jednotek OTU

- **Výsledky analýzy PCA a LDA**

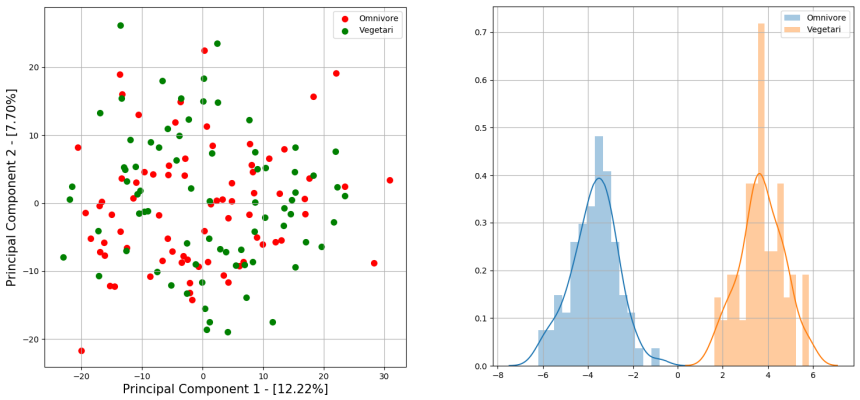


Obrázek 8.1: Výsledek *PCA* analýzy *(vlevo)* a *LDA* analýzy *(vpravo)* atributu *diet_type* na úrovni *L6* při původním rozsahu celkového počtu jednotek *OTU* ve vzorcích

Experimenty omezeném rozsahu jednotek OTU

Další možností byla užší specifikace vstupní datové sady a rozdělení datové sady na menší intervaly podle celkového počtu jednotek *OTU* ve vzorku. Z datové sady byly vybrány pouze vzorky s rozmezím celkového počtu jednotek *OTU* mezi *20000* a *25000*. Na tomto intervalu se oproti původní datové sadě nachází *738* vzorků, z toho *653* masožravců a *75* vegetariánů. Stojí za zmínku, že i při takto specifikovaném výběru se přesnost klasifikátorů nijak výrazněji nezměnila, za pomocí *LDA* analýzy je ale možné dané třídy od sebe vzájemně separovat.

- **Výsledky analýzy PCA a LDA**



Obrázek 8.2: Výsledek *PCA* analýzy *(vlevo)* a *LDA* analýzy *(vpravo)* atributu *diet_type* na úrovni *L6* při omezeném rozsahu celkového počtu jednotek *OTU* ve vzorcích

8.6 Diskuze výsledků

Výsledky nasvědčují, že data jsou obecně lépe separovatelná na nižších taxonomických úrovních, a to *L5* a *L6*, na vyšších stupních rozdílnost mezi třídami zaniká. Z použitých algoritmů bylo poměrně těžké rozlišit, který klasifikátor si vedl nejlépe. Při testování modely i při využití křížové validace vykazovaly poměrně vysoké odchylky v jednotlivých měřeních, což může být odrazem velkého množství rysů v tabulce OTU s původně nulovou hodnotou.V případě klasifikace i jiných životních atributů se výsledky zásadně nezměnily, totéž platilo i při selekci pouze omezeného intervalu celkového počtu jednotek OTU ve vzorcích. Největší pozornost byla věnována rozdílným stravovacím návykům, které by z dostupné literatury měly nejvíce reflektovat zastoupení bakterií střevní mikroflóry.

Konkrétní jev LDA analýzy, který můžeme vidět v grafu 8.2 se stejně tak opakuje i u jiných atributů než pouze u zmiňovaných stravovacích návyků. Z toho důvodu usuzuji, že tento výsledek nemá přímou souvislost s danými atributy, ale spíše se jedná o zkreslení datové sady či nízký počet testovacích vzorků, který toto rozdělení umožňuje. I tak je možné říci, že značný vliv pro budoucí klasifikaci může mít původní hloubka sekvenování vzorku, která má za následek větší počet taxonů zastoupených ve vzorku. V tomto případě připadají v úvahu dvě možná řešení. Buď vzít pouze vzorky s podobným celkovým množstvím jednotek OTU a pro tyto vzorky vytvořit zvlášť jednotlivé klasifikátory, a to nejlépe s využitím analýzy LDA, nebo vzorky s původně vysokým počtem rozdílných taxonů zpětně rarefakovat na nižší počet jednotek OTU ve vzorku.

Výsledné modely implementovaného prediktoru zakládají na klasifikátoru SVM s jádrovou funkcí *rbf* a parametrem $C = 1$. Tyto modely jsou k dispozici ve složce `models/finals/`. Vstupní vzorky jsou k dispozici ve složce `biom/`.

Kapitola 9

Závěr

Výsledkem práce je implementovaný nástroj pro predikci životních atributů. Pomocí tohoto nástroje byla odzkoušena predikce životních atributů určující stravovací návyky, potravinové alergie na lepek či laktózu, bmi a věk pacienta. V případě klasifikace byly vzorky rozděleny vždy do dvou tříd. Experimentováno bylo s klasifikátory založenými na různých algoritmech strojového učení. Nástroj implementuje postupně metody k nejbližších sousedů, náhodný les a metodu podpůrných vektorů. Experimenty zahrnovaly jak různé nastavení parametrů klasifikátorů, tak vyhodnocení na různých taxonomických úrovních. V neposlední řadě bylo taktéž odzkoušeno více životních atributů.

Bohužel na žádné z těchto úrovní nebylo možné prokázat významnější separabilitu daných tříd a dosažené výsledky nenasvědčují že jsou data oddělitelná tímto způsobem. Nejlepší výsledky predikce byly s 61% přesností. Průměr predikovaných hodnot se ale pohybuje pouze lehce nad *54%*. Data byla taktéž zkoumána za pomocí analýz *PCA* a *LDA*. Ani tyto analýzy nenasvědčují, že jsou data od sebe vzájemně separovatelná. K rozdělení došlo pouze při omezeném výběru vzorků z původní datové sady na taxonomické úrovni *L6*. Další možností pokračování by tak mohlo být vytvoření více různých klasifikátorů s využitím znalostí analýzy *LDA* nebo případně natrénování klasifikátoru na jiných datových sadách s mírnou úpravou implementovaného prediktoru.

Literatura

[1] Agrawal, R.: K-Nearest Neighbor for Uncertain Data. 2014.
URL https://www.semanticscholar.org/paper/K-Nearest-Neighbor-for-Uncertain-Data-Agrawal/67a7e283eced2d6e2275de0ef0bf92c0d2364c40

[2] Anděl, J.: *Matematická statistika*. SNTL - Nakladatelství technické literatury Praha, druhé vydání, 1985.

[3] Balvočiūtė, M.; Huson, D. H.: SILVA, RDP, Greengenes, NCBI and OTT — how do these taxonomies compare? *BMC Genomics*, ročník 18, č. 2, Mar 2017: str. 114, ISSN 1471-2164, doi:10.1186/s12864-017-3501-4.
URL https://doi.org/10.1186/s12864-017-3501-4

[4] David, L. A.; Maurice, C. F.; Carmody, R. N.; aj.: Diet rapidly and reproducibly alters the human gut microbiome. *Nature*, ročník 505, Dec 2013: s. 559 EP –.
URL https://doi.org/10.1038/nature12820

[5] Gerharz, T.: Identifikace mikroorganizmů pomocí state-of-the-art techniky. *CHEMAGAZÍN*, 2016: s. 21–23.
URL
http://www.chemagazin.cz/userdata/chemagazin_2010/file/chxx_1_cl6.pdf

[6] Gloor, G. B.; Macklaim, J. M.; Pawlowsky-Glahn, V.; aj.: Microbiome Datasets Are Compositional: And This Is Not Optional. *Front Microbiol*, ročník 8, Nov 2017: s. 2224–2224, ISSN 1664-302X, doi:10.3389/fmicb.2017.02224, 29187837[pmid].
URL https://www.ncbi.nlm.nih.gov/pubmed/29187837

[7] Gron, A.: *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., první vydání, 2017, ISBN 1491962291, 9781491962299.

[8] Hansen, L. B. S.: A low-gluten diet induces changes in the intestinal microbiome of healthy Danish adults. *Nat Commun*, ročník 9, č. 1, Nov 2018: s. 4630–4630, ISSN 2041-1723, doi:10.1038/s41467-018-07019-x, pMC6234216[pmcid].
URL https://www.ncbi.nlm.nih.gov/pubmed/30425247

[9] Hejret, T.: Rozhodovací stromy a jejich aplikace při analýze dat. 2015, technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, Katedra informatiky, 49s. Vedoucí diplomové práce: Ing. Petr Gajdoš, Ph.D.
URL https://dspace.vsb.cz/bitstream/handle/10084/108605/HEJ124_FEI_N2647_2612T025_2015.pdf

[10] Hodkinson, B. P.; Grice, E. A.: Next-Generation Sequencing: A Review of Technologies and Tools for Wound Microbiome Research. *Adv Wound Care (New Rochelle)*, ročník 4, č. 1, Jan 2015: s. 50–58, ISSN 2162-1918, doi:10.1089/wound.2014.0542[PII], 25566414[pmid].
URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4281878/

[11] Honzík, P.: *Strojové učení*. 2006, 85s. skripta - FEKT Vysokého učení technického v Brně, [Online; navštíveno 26.04.2019].
URL http://midas.uamt.feec.vutbr.cz/STU/Lectures/Honzik%20-%20Strojove_uceni_S.pdf

[12] Karlsson, F.; Tremaroli, V.; Nielsen, J.; aj.: Assessing the Human Gut Microbiota in Metabolic Diseases. *Diabetes*, ročník 62, č. 10, 2013: s. 3341–3349, ISSN 0012-1797, doi:10.2337/db13-0844,
http://diabetes.diabetesjournals.org/content/62/10/3341.full.pdf.
URL http://diabetes.diabetesjournals.org/content/62/10/3341

[13] Knights, D.: Microbiome Discovery. Online lectures held at Department of Computer Science and Engineering, Biotechnology Institute - University of Minnesota.
URL
https://www.youtube.com/playlist?list=PLOPiWVjg6aTzsA53N19YqJQeZpSCH9QPc

[14] Martinez, A. M.; Kak, A. C.: PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 23, č. 2, Feb 2001: s. 228–233, ISSN 0162-8828, doi:10.1109/34.908974.
URL http://www2.ece.ohio-state.edu/~aleix/pami01.pdf

[15] Meloun, M.: *Interaktivní statistická analýza dat*. Nakladatelství Karolinum, třetí vydání, 2012, ISBN 978-80-246-2173-9.

[16] Mullin, E.: The bacteria in your gut may reveal your true age. *Science Magazine*, Jan 2019.
URL https://www.sciencemag.org/news/2019/01/bacteria-your-gut-may-reveal-your-true-age

[17] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.

[18] Pelliccia, D.: Classification of NIR spectra by Linear Discriminant Analysis in Python. [Online; navštíveno 23.04.2019].
URL https://www.idtools.com.au/classification-nir-spectra-linear-discriminant-analysis-python/

[19] Tierney, B.: Random Forest Machine Learning in R, Python and SQL. 2018.
URL https://blog.toadworld.com/2018/08/31/random-forest-machine-learning-in-r-python-and-sql-part-1

[20] Uhlík, O.; Strejček, M.; Hroudová, M.; aj.: Identifikace a charakterizace bakterií s bioremediačním potenciálem – od kultivace k metagenomice. *Chemické Listy*, ročník 107, 2013: str. 614–622.
URL http://www.chemicke-listy.cz/docs/full/2013_08_614-622.pdf

[21] Upasana, V.: *How to handle Imbalanced Classification Problems in machine learning?* [Online; navštíveno 22.03.2019].
URL https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

[22] Weiss, S.: Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome*, ročník 5, č. 1, Mar 2017: s. 27–27, ISSN 2049-2618, doi:10.1186/s40168-017-0237-y.
URL https://www.ncbi.nlm.nih.gov/pubmed/28253908

[23] Wikipedia: *Coefficient of determination*. [Online; navštíveno 22.03.2019].
URL https://en.wikipedia.org/wiki/Coefficient_of_determination

[24] Wikipedia: *Taxonomic rank*. [Online; navštíveno 22.03.2019].
URL https://en.wikipedia.org/wiki/Taxonomic_rank

[25] Yun, Y.; Kim, H.-N.; Kim, S. E.; aj.: Comparative analysis of gut microbiota associated with body mass index in a large Korean cohort. *BMC Microbiology*, ročník 17, č. 1, Jul 2017: str. 151, ISSN 1471-2180, doi:10.1186/s12866-017-1052-0.
URL https://doi.org/10.1186/s12866-017-1052-0

[26] Čejková, D.; Šmajs, D.: Projekt lidského mikrobiomu. *Universitas*, 2010: s. 24–28.
URL https://journals.muni.cz/universitas/article/view/1472/1099

Příloha A

Obsah přiloženého CD

- `src/`
  - `mod_classifier/` - Implementace modelů třídních klasifikátorů.
  - `mod_regression/` - Implementace modelů regresních klasifikátorů.
  - `analysis.py` - Implementace tříd *PCA* a *LDA*.
  - `classifier.py` - Implementace třídy *Classifier*.
  - `clr.py` - Implementace třídy *CLR*.
  - `dataset.py` - Implementace třídy *Dataset*.
  - `otu.py` - Implementace třídy *OTU*.
  - `preprocess.py` - Implementace třídy *Preprocess*.
  - `t_test.py` - Implementace třídy *T_test*.
  - `metagen-predictor.py` - Hlavní skript klasifikátoru.
- `data/`
  - `scripts/` - Skripty použité pro získání a zpracování datové sady.
  - `biom/` - Vzorové soubory ve formátu *BIOM* určené pro predikci.
  - `attributes_f.csv` - Atributy k datové sadě.
  - `{L2-L6}_taxa_a.csv` - Základní datové sady na tax. úrovních *L2* až *L6*.
  - `{L2-L6}_taxa_f.csv` - Filtrované datové sady na tax. úrovních *L2* až *L6*.
- `models/` - Vzorové natrénované modely společně s grafy analýz PCA a LDA.
- `thesis/`
  - `src/` - Zdrojové soubory technické zprávy.
  - `xkubic39-ibt.pdf` - Bakalářská práce ve formátu *PDF*.

Příloha B

Příklad spuštění prediktoru

Kompletní nápovědu programu `metagen-preictor.py` popisující všechny vstupní parametry lze zobrazit příkazem:

```
$ metagen-predictor.py --help
```

- Příklad pro **natrénování** modelu klasifikátoru pro atribut *diet_type* na taxonomické úrovni *L6* s filtrací tabulek *OTU* v intervalu *5000* až *40000* s vyvažováním metodou *podvzorkování* a využitím metody *t-test* pro redukci mnoharozměrných dat:

  ```
  $ metagen-predictor.py train \
     --model-path ./models/L6 \
     --taxa-level L6 \
     --attributes diet_type \
     --algorithms svm \
     --otu-count-range 5000,40000 \
     --imbalance-method undersampling \
     --t-test
  ```

- Příklad pro **otestování** modelu klasifikátoru pro atribut *diet_type* na taxonomické úrovni *L6*:

  ```
  $ metagen-predictor.py test \
     --model-path ./models/L6 \
     --taxa-level L6 \
     --attributes diet_type
  ```

- Příklad pro **predikci** atributu *diet_type* na vzorku ze souboru `./biom/ERR1072639.biom` na taxonomické úrovni *L6*:

  ```
  $ metagen-predictor.py predict \
     --input-file ./data/biom/ERR1072639.biom \
     --model-path models/final \
     --taxa-level L6 \
     --attributes diet_type
  ```