Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Computer science

Master's thesis

Visualization and analysis of patients digital phenotypes

David Kolečkář

Supervisor: doc. Ing. Daniel Novák, Ph.D.

Field of study: Open Informatics
Subfield: Bioinformatics

Prague, May 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kolečkář** Jméno: **David** Osobní číslo: **420230**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Bioinformatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Vizualizace a zpracování dat chronických pacientů

Název diplomové práce anglicky:

Visualization and analysis of patients digital phenotypes

Pokyny pro vypracování:

Given the spread of mobile technology use which is reflected by the lived experiences of people in their natural environments, it should be possible to leverage it to develop precise and personalized disease phenotypes and markers to diagnose, monitor, and treat illnesses
1) Familiarize yourself with the issue of phenotype analysis and personalized medicine. Focus on the techniques recently emerged, among others topological data analysis , temporal multivariate longitudinal data clustering or latent variable models or semi-supervised deep clustering.
2) Perform exploratory analysis and apply methods for data visualization.
3) Compute digital phenotypes of patients applying advanced clustering methods.
4) Perform evaluation using measures as mutual information, rand index, Calinski-Harabasz or Davies-Bouldin score

Seznam doporučené literatury:

1. Torous, Kiang MV, Lorme J, Onnela JP. JMIR Ment Health. 2016;3(2)
2. Onnela JP, Rauch SL. H Neuropsychopharmacology, 41(7):1691-6. 2016
3. Benoit J, Onyeaka H, Keshavan M, Torous J. Harv Rev Psychiatry. 2020

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Daniel Novák, Ph.D. Analýza a interpretace biomedicínských dat FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **09.03.2023** Termín odevzdání diplomové práce: **26.05.2023**

Platnost zadání diplomové práce: **16.02.2025**

doc. Ing. Daniel Novák, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.	
Datum převzetí zadání	Podpis studenta

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act

Prague, on 25. May 2023

.....
David Kolečkář

Acknowledgments

I would like to express my heartfelt thanks, first and foremost, to the supervisor of this thesis, doc. Ing. Daniel Novák, Ph.D., for his time, unwavering assistance, and the wonderfully prepared Pu'er tea served during our numerous meetings. I would also like to extend my gratitude to Mgr. Tomáš Sieger, Ph.D., for his invaluable experience and practical guidance in designing the experiments, as well as to Ing. Václav Burda and Ing. Jindřich Prokop for their insights into the smoking cessation application. I would like to extend my sincere thanks to RNDr. Milan Straka, Ph.D., who has sparked within me a profound passion for deep machine learning. Last but not least, I would like to thank myself for the hard work and perseverance.

Abstract

This thesis aims at analyzing chronic patients data, trying to identify subgroups in the population for which a more personalized treatment can be provided, using state-of-the-art unsupervised clustering and deep learning methods. Focusing further on the patients with chronic tobacco addiction, overview of the latest m-health and deep phenotyping approaches is given. Suitable dataset is created from the database of smoking cessation smartphone application with real patients data, of cardinality of approximately 5000 patients with 20 measured, mainly sociodemographic, features. Issues emerged by working with dataset with mixed – numerical and categorical – variables are solved. It is done in rather innovative manner, proposing new technique of training Entity Embeddings for special case of nominal variables. Latest dimensionality reduction methods are discussed, namely the Autoencoders and Variational Autoencoders. Later the same is done for the clustering algorithms. Afterward, for the created datasets exploratory analysis is carried on, yielding valuable descriptive and graphical information about the patients population. Finally the patients are clustered, using algorithms. From the emerged patients clusters are estimated the digital phenotypes. Depending on the numerous hyperparameters 7 to 8 well defined clusters, phenotypes, were identified in the patients population. These results enable future personalized therapy improvements and provide valuable feedback for the patients data collection process.

Keywords: deep phenotyping, smoking cessation app, Autoencoders, unsupervised learning, Entity Embeddings

Abstrakt

Tato diplomová práce si klade za cíl analyzovat data pacientů s chronickými onemocněními a identifikovat podskupiny jejich populace, pro které lze poskytnout více personalizovanou léčbu pomocí nejmodernějších metod shlukování a hlubokého učení. Zaměřuje se dále na pacienty s chronickou závislostí na tabáku a poskytuje přehled nejnovějších přístupů v oblasti mobilního zdravotnictví (m-health) a hlubokého fenotypování. Pro tyto účely byl vytvořen vhodný datový soubor z databáze mobilní aplikace pro odvykání kouření s reálnými daty pacientů, o velikosti přibližně 5000 pacientů a 20 měřených, převážně sociodemografických, charakteristik. Dále jsou v rámci studie řešeny problémy spojené s prací s daty obsahujícími smíšené - numerické a kategoriální - proměnné jsou řešeny. Následně je představena technika pro získání vhodných reprezentací speciálního případu nominálních proměnných, pomocí trénování neuronových sítí. Jsou diskutovány nejnovější metody shlukování a redukce dimenzionality jako jsou Autoenkodéry a Variační Autoenkodéry. Poté je provedena explorativní analýza vytvořených datových sad, jejímž výsledkem jsou cenné popisné a grafické statistické informace o populaci závislých na nikotinu. Nakonec je populace rozdělena do shluků shlukovacími algoritmy, což umožňuje odhad digitálních fenotypů uvnitř jednotlivých shluků pacientů. V závislosti na různých hyperparametrech bylo identifikováno 7 až 8 dobře definovaných shluků, fenotypů, v populaci. Zjištěné digitální fenotypy umožňují budoucí zlepšení personalizované terapie a také poskytují hodnotnou zpětnou vazbu pro budoucí sběr dat pacientů skrz mobilní aplikaci.

Klíčová slova: digitální fenotypizace aplikace na odvykání kouření, autoenkodery, učení bez učitele, Entity Embeddings

Used shortcuts

api – application interface

app – application

dof – degrees of freedom

pca – principal component analysis

svd – singular value decomposition

mds – multi-dimensional scaling

ae – autoencoder

vae – variational autoencoder

tf – TensorFlow (deep learning framework)

json - JavaScript Object Notation (data format)

NaN – Not a number

y.o. - years old

nlp – natural language processing

dimred – dimensionality reduction

Table of Figures

Index of Tables

Table of Contents

Chapter 1

Introduction

This chapter explains the motivation behind writting this thesis. Following chapters present the actual context of personalized medicine and emerged techniques leveraging the various machine learning methods to identify subgroups in the chronic patients population on the chosen subset of measurable patients population features.

Motivation

In an ideal world, patients in general, including those with chronic conditions, would receive personalized medical treatment. To achieve this, it is necessary to create an apparatus that categorizes patients based on a measurable set of features. While such a task may vary depending on the specific disease domains, there are commonalities across various domains that can be utilized in developing such algorithm. We would like to inspect such possibilities, creating a suitable analysis pipeline and subsequently test proposed approach on real chronic patients data. If successful, such personalized and more effective therapy could have strong health and social benefits. Using machine learning methods to create digital phenotypes for some population of chronic patients is quite a novel approach and with smartphones and other smart devices being ubiquitous in modern society, it would be unwise not to make use of such platform.

Outline

The goal of this work is to create digital phenotypes for chosen suitable subset of real chronic patients. Once the specific domain is chosen, it is important to conduct a thorough research to determine the most relevant features of the chronic patient population. These features will be used to create the resulting digital phenotypes. Next, it is important to obtain real patient data and establish a preprocessing pipeline to generate a dataset that is appropriate for subsequent analysis. After creating the dataset, an exploratory analysis will be conducted to obtain descriptive statistics and visualizations. This analysis aims to gain a comprehensive understanding of the chronic patient population, enabling more informed and data-based

decisions to guide the final clustering steps in a meaningful direction. Since the dataset will be of high dimensionality, it is necessary to employ dimensionality reduction methods for inspecting the dataset and visualizing the clustering results. Additionally, a wide range of state-of-the-art clustering and dimensionality reduction techniques will be employed to determine the optimal separation of the chronic nicotine users' population. Various metrics will be utilized to evaluate the performance of the clustering methods. Finally the emerged phenotypes for the found chronic patients groups will be created.

Contribution

This work creates - to our best knowledge – one of the first more complex digital phenotyping pipelines applied on a real (mainly) Czech patients with chronic nicotine addiction, using the latest unsupervised machine learning methods. Our results can be further utilized by medical experts for targeting the right patients subgroups, creating more personalized therapy. Also, the exploratory analysis provides many interesting statistics about chronic smokers. One of the outputs of this work is a clean and well-documented dataset of approximately *five thousand* patients and *twenty* measured features. This work uses the deep generative models for dimensionality reduction, namely the Autoencoders and Variational Autoencoders, which is a rather newer approach. Possibly, a new way of designing the encoding process for a specific subgroup of nominal variables, whose values are subsets of the set of options patient can choose from (e.g. by filling out a questionnaire), is presented. This encoding procedure could be described as: “learning the Entity Embeddings by neural networks, formulated as a multi-label classification task”.

Chapter 2

M-health and Personalized medicine

This chapter starts by summarizing the current trends in m-health and personalized medicine, focusing then more on the smartphone therapy applications for chronic patients. In the end an overview of nicotine cessation apps is given in general. Further, a particular smartphone *app* is introduced, from which the chronic nicotine users data for further analysis were obtained. Also, a *digital phenotype* is defined. The types of profiling usually used for cigarette smokers and nicotine users in the literature is presented as well.

2.1 Mobile apps for chronic diseases management

Firstly, an overview of applications targeting chronic patients in general is given, followed by an overview of smoking cessation apps. Mobile phones are one of the most abundant digital platforms in today's society. To make us of it's huge potential for gathering patients data and for providing therapy seems very reasonable, if tasks like providing more precise and personalized medical treatment wants to be tackled. Long term health-management for patients with chronic diseases was studied, and it was found, that using smartphones is a promising option *[1]*. The retrospective analysis of users of a medication adherence management mobile app revealed a positive trend in maintaining optimal medication adherence over time *[2]*. The mobile health tracking apps targeting patients with chronic conditions were studied from an architectural point of view in *[3]* and it was find out, that the apps should be highly usable to motivate chronic patients to continue in the virtual therapy. A difference in motivation regarding seriousness of the quitting attempts was found between patients using Android and iOS *[4]*. The difference in abstinence duration for patients using decision-apps and those using information-only apps, was studied in *[5]*. The quality of provided therapy mainly for apps for pain management was studied in *[6]* and it was found out, that one of the reasons behind a poor quality of some apps could be the lack of a regulatory body assessing their qualities. Overall challenges in the collection and analysis of digital phenotyping data were discussed more in depth in *[7]*. A large review of smartphone applications for smoking cessation was conducted by *[8]*. Effectiveness of mobile apps for smoking cessation and the various reasons for patients could be quitting the

therapy was studied in *[9]*. The long-term effectiveness, of one of the main smoking cessation apps on the marked, was described studied in *[10]*. These studies and their results only punctuate the importance of using the mobile technologies for personalized medicine and shows the growing trend in m-health and using smartphones for therapy reasons, as a medical instrument.

2.2 Digital Phenotype

The main goal Digital Phenotyping is to collect high-quality smartphone data and to create the necessary methodology to make use of such data, as first described in *[11]*. This field of study was, from the year 2016 when the term emerged, intensively studied in the literature. Profound study of opportunities and challenges presented by the collection and analysis of digital phenotyping data was made by *[12]*. A more specialized approach towards precision health and patient-centered care given in *[13]*, proposing term *Digital Twin.* Digital Phenotyping quality and safety related issues are inspected into more detail in *[14]*. The ethical point of view then in *[15]*. It is appropriate to ask such questions about ethics and discuss safety related issues. Although Digital Phenotyping is mainly a data-driven approach, it is important not to forget, that the data still represent real people. Also, the numerous studies prove, that the digital footprint could be used for something good, not just for a marketing purposes.

The data gathered from the smartphone (or any other personal mobile platform) can be divided further into 2 groups: *passive* and *active*. Passive data are mainly some smartphone-sensor generated data, created without patients input. This thesis mainly works with the active data, generated by patients directly interacting with the device. No biomarkers or any additional biochemical information are at disposal, even though having such features would be beneficial.

2.3 Smokers profile

Profiling smokers in literature is done in many ways. The most primitive one is to categorize the patients into usually three categories by the number of smoked cigarettes per day. The categories are following: light, moderate and heavy smokers, smoking 1 to 10, 11 to 19 and more than 20 cigarettes per day, respectively. Other categories could be *'Never-smokers', 'Daily smokers', etc.* based on the frequency or previous duration patients used to smoke. More profound profiling for some smokers subgroups was done by *[16]*.

2.4 Used smoking cessation app structure

Data are created by patients interacting directly or indirectly with the smartphone application. The application offers virtual therapy sessions either each day or periodically in later application stages. Patient is encouraged to interact with the application during each session either by answering questions about current physical or psychological state or by chatting with the application . These interactions generate the features used for further analysis. The digital trace patient leaves in the application is rather large, so only the major sociodemographic and some other features are later chosen. There are two versions of the *app*, a free trial and a paid version. The free version doesn't let user further than *EE* phase, defined in *Table 1*, thus splitting the emerged datasets into two parts.

2.4.1 Therapy phases

The therapy phase represents patient's progress in the therapy. There are multiple consecutive therapy phases, each one with exactly defined number of sessions, one per day. User must finish current therapy phase to continue to the next one. The construction of the sessions and their content were designed by medical experts. *Table 1* and the flow graph on *Figure 1* describe the therapy phases.

Table 1: Therapy phases and their meaning.

Phase name	Phase description
START	Patient created account, but didn't start the therapy yet.
EE	Patient started the therapy.
	This phase consists of 10 therapy sessions (1 session per day). Those sessions however don't have to be completed continuously, there is no upper day limit. Patient can also choose to complete all of the 10 '*EE*' sessions in 1 day, which is referred to as '*bujon*'.
EQ	The day patient stops smoking. Patient gets stuck in this phase otherwise.
FU	Intensive follow-up phase subsequent to cessation, consisting of 21 sessions.
WR	Maintenance phase, consisting of 70 consecutive days of therapy.
FIN	User finished all phases, therapy ends.

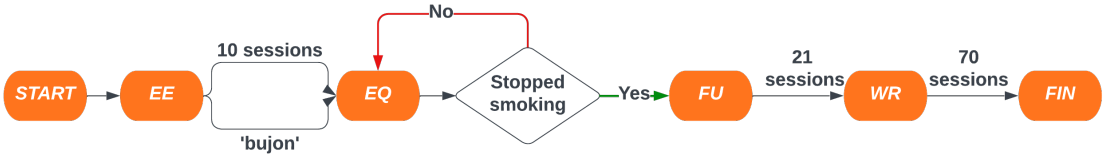

Figure 1: Therapy phases flow diagram. Image shows possible therapy phases used in the used smoking cessation app, duration of the phase is represented by the transition to the subsequent phase.

Chapter 3

Methods and algorithms

This chapter presents the statistical methods and machine learning algorithms used during the exploratory analysis and in the subsequent clustering and evaluation steps performed in order to create the patients digital phenotypes. Firstly, general overview of method categories suitable for this task is presented. In the following parts, the methods are discussed in chronological order, as used during the experiments pipeline. Starting by exploratory analysis and correlations tests. Followed by dimensionality reduction techniques, clustering methods and clustering performance evaluation methods. Special section is dedicated to deep generative models, namely the Autoencoders and their regularized variant Variational Autoencoders. However, these models are not used as generative models in this work, but for dimensionality reduction. In the end of this chapter, possible ways of encoding categorical variables are discussed. New[1] approach of training deep embeddings for categorical nominal variables, with values as subsets of a set *{1, …, n}* (results from a questionnaire with *n* options), is presented.

3.1 Digital phenotyping approaches

This section briefly discusses possible approaches to analyzing chronic patients data and creation of theirs digital phenotypes. There are many possible ways to approach the task of digital phenotyping, mainly depending on type of measured features for the patients population. The data this thesis works with does not contain implicit time information, thus methods using time series in some way would not be applicable. The data collected from the cessation smartphone app are mainly sociodemographic active data provided directly by the patients themselves. Main preprocessing problem will be working with the mixed dataset, transforming it into suitable numerical form. Only then, standard clustering techniques can be used.

1

1 Learning E*ntity Embeddings* by neural networks for L*abel-encoded* categorical variables is common approach, but not defining the optimization task as multi-label classification in order to caption subset similarities.

3.2 Exploratory analysis

This part presents the statistical methods used during the exploratory analysis. These methods provide means for obtaining both descriptive and graphical statistics. Firstly, methods from univariate analysis are described and consequently the methods used in bivariate and multivariate analysis are presented. Univariate analysis is the introductory step in order to understand the dataset, mainly by visualizing variables distributions, calculating central tendencies and measuring variability in the population. For continuous variables arithmetic mean, standard deviation, skewness and kurtosis is calculated. For categorical variables, either ratios are calculated, or median in case of categorical ordinal. Used graphical methods are mainly histograms and bar charts.

Following section refreshes statistical methods used for testing pairwise variables relationship. Main focus is on pairwise linear correlations and testing differences between expected values for some variable between two independent populations. This step is not only desirable for understanding the data, but also a necessary preprocessing step prior to the dimensionality reduction and clustering algorithms. Multicollinearity would skew the results. Simply put, two strongly correlated dimensions carry only one piece of information, but would be 'counted' twice.

3.2.1 Correlations

For testing pairwise variables relations following methods will be used: Pearson correlation coefficient (*Pearson's r*), Spearman's rank correlation coefficient (*Spearman's* ρ), Kendall rank correlation coefficient (*Kendall's* τ) and Chi-square test of independence. Detailed formulas given in *[17]*. *Table 2* shows used pairwise independence tests and correlation coefficients, coefficient ranges and their meaning.

Table 2: Correlation coefficients and pairwise independence tests statistic, their ranges and interpretations .

Variable type	Correlation coefficient	Result range	Result meaning
numeric	*Pearson's r*	*<0, 1>*	0 = uncorrelated 1 = highly correlated
ordinal	*Spearman's* ρ	*<-1, 1>*	1 = positive monotonic relationship -1 = negative monotonic relationship
	Kendall's τ	*<-1, 1>*	1 = ranking of variable A is the same as ranking of variable B, -1 = perfect inversion
binary	χ^2 *(not a coefficient, but value of the statistic)*	*<0, inf>*	Depends on the significance level the test is performed. But for two binary variables and *dof=1* the critical value is ***3.841***.

3.2.2 Comparing two independent population means

Using advanced clustering and machine learning algorithms directly could be tempting. But it would be better, if the relationship between some variables like '*age*' and '*therapy success*' could have been found by common statistical methods. The results would be much more explicable, than those obtained from some advanced method, because interpreting clustering results could be substantially intriguing.

First the tests for testing normality and homoscedasticity are listed then the methods for testing differences between two population means for some variable. *Shapiro-Wilk test* tests the null hypothesis $H_{0,}$ that data was drawn from a normal distribution. *Levene* or *Bartlett tests* are used to asses equality of variances in two populations for some variable. Finally, *ANOVA* and it's non-parametric counterpart *Kruskal-Wallis test by ranks* will be used, for testing whether two (or more) groups statistically significantly differ for some independent variable. *[17]*

3.3 Dimensionality reduction

Dimensionality reduction is an essential preprocessing step when dealing with high-dimensional datasets. It enables exploring potential patterns within the data before applying clustering algorithms and facilitates visualization of the results. Moreover, dimensionality reduction enhances the effectiveness of clustering algorithms, by eliminating irrelevant or redundant features, it can improve clustering performance. Ideally, if the projections show clustered data, the clustering algorithms should find them as well. In order to maximize probability of finding some real structure in the data in two dimensions, plethora of algorithms from maximally orthogonal algorithm 'families' will be used, namely: *Principal Component Analysis [18], Isomap [19], t-SNE [20], UMAP [21], Multi-Dimensional Scaling[22], Autoencoder[23] and Variational Autoencoder [24]*.

Following section does not aim to provide exhaustive comparison of dimensionality reduction techniques such as *[25]*, but rather discuss when is suitable to use the specific algorithm. Also to present the various hyper-parameters that have to be tuned. Regarding the hyper-parameters, they can, and usually do, vary depending on the used implementation. Many new implementations of algorithms described in the original papers tend to upgrade the performance by some changes in the original algorithm or by solving some niche use-case, that was not taken into account in the original paper. Since choosing the hyper-parameters can be sometimes non-intuitive, there have been many efforts to automate the hyper-parameter selection or to provide reasonable set of defaults, for example for t-SNE in *[26]*. Also, all the used algorithms require numerical input, or some metric that can handle non-numerical data must be provided. Usually

the default metric is considered to be the *L2 norm* (Euclidean distance). So, if the dataset contains mixed variables, either suitable metric like G*ower's distance [27]* must be provided or all non-numerical variables must be transformed into numerical.

3.3.1 Hyper-parameters

For example, the number of possible hyper-parameters of *scikit-learn* Isomap implementation is rather large ~11 (leaving out performance-related ones such as the number of threads to use). Aim of this section, is not a detailed description of each algorithm. However good understanding of the hyper-parameters is necessary to be able to navigate the experiments in the right direction. For the each algorithm a reasonable default is used. However some parameters seem to have more serious effect on the algorithm run than others *[28]*, so for those 'more important ones' some kind of search should be usually performed. Also one thing is a set of parameters as given in the original paper and second thing is the set of parameters and hyper-parameters offered by the specific implementation, which tends to be usually non trivially larger. *Table 3* summarizes *dimred* algorithms used later in this work, number of their major hyper-parameters and their basic meaning. The projection dimension is called differently for each algorithm. For example, for *PCA* it is the chosen number of the first '*k*' principal components, for *(V)AE* it is the latent dimension usually denoted '*z*'. *Table 4* presents whether for each run, the algorithm gives the same results.

Table 3: Used dimensionality reduction algorithms and their hyper-parameters. Metric used by the algorithms are not counted as a hyper-parameters in this table, as well as projection dimension.

Algorithm	Parameters	Notes on parameters
PCA	0	optionally the parameters regarding *SVD* solver
MDS	0	Mainly internally used *SMACOF algorithm parameters*
Isomap	1	Number of neighbors
t-SNE	2	perplexity, number of iterations
UMAP	2	Number of neighbors, minimal distance for point to be core
AE / VAE	very high	the number and order of hidden layers, layers sizes and activations, normalization elements, training parameters, optimizer, ...

Table 4: Stochasticity of dimensinality reduction algorithms results.

PCA	MDS	Isomap	t-SNE	UMAP	AE/VAE
deterministic	stochastic	deterministic	stochastic	stochastic	stochastic

3.3.2 Practical notes for some algorithms

Following section tries to discuss the algorithms from more practical point of view, because in order to obtain reasonable results, can be quite finicky task and usually require some kind of a hands-on experience. Not mentioning that algorithm interface usually differs for different implementations. This should rather create some practical intuition for using the algorithms. An intuition that is leveraged in the subsequent experiments.

PCA should be used only for standardized numerical features, however *[29]* presents a way how to use *PCA* for mixed a dataset. *PCA* should be used, when many of the variables are highly correlated with each other, to reduce their number to an independent set. If found, during the bivariate analysis, that the numerical variables are not correlated, it could be expected that that *PCA* will not be able to explain much of the variance in the dataset by using fewer dimensions.

According to experiments done in *[28]*, studying how t-SNE behaves on simple cases, important algorithm behavior was observed:

- Keep the perplexity in range from 5 to 50, as given in *[20]*.
- Different datasets require different number of iterations to converge, but usually 5000 should be sufficient.
- Probably relative sizes of clusters are not preserved, the resulting clusters sizes are not relevant, as t-SNE tends to contract sparse clusters and expand dense ones.
- Distances between the resulting clusters might not be relevant as well.
- Topological information can be sometimes retrieved by plotting the data with various perplexities.

Perplexity could be interpreted as the number of nearest neighbors that are considered during the algorithm run in the original space, that implies that with increasing perplexity more of a global data structure is captured, of course for the given metric that is used.

3.4 Autoencoders

This section describes the deep generative models, a neural networks family used and implemented in this thesis. However it is used not as a generative model, but for dimensionality reduction, using only the encoder part of the trained network. Autoencoder as described in *[23]* can be considered a latent-variable model as described in*[30]*. The autoencoder family, as, to date, is substantially large, the main branches being:

- Denoising Autoencoders *[31]*,
- Sparse Autoencoders *[32]*,

- Variational Autoencoders *[24]*
- Contractive Autoencoders *[33]*

This thesis further works with the “plain” Autoencoder as described in *[23]* and with the Variational Autoencoder as described in *[24]*.

3.4.1 Autoencoder architecture

Autoencoders are usually depicted as an ‘hourglass’ shape architecture, however the input layer does not have to be larger than the hidden layer, which doesn’t hold for the implementation in this thesis neither. Usually much deeper networks or networks with higher capacity are used. For purposes of this thesis is however sufficient quite a shallow network, both encoder and decoder having just 3 hidden layers, of sizes 128, 64 and 32 with the last layer having only 2 dimensions, see *Figure 2.* This encoder structure generates what is referred to as the *latent space [34]* – the embedding dimension, the bottleneck. Architecture of the decoder is mirroring the encoder. In the network depicted in *Figure 2* and *Figure 3* is the latent dimension equal to 2, because the autoencoder was used as a tool for projecting the data it was trained on, into 2 dimensions for visualization. That is the main idea of using the autoencoder as dimensionality reduction technique. First the model is trained on the dataset to convergence. It does not matter if it overfits, and it most probably does, since no regularization is used and the number of data could be small. This would matter if the model would be used as a generative one, but for dimensionality reduction purposes that does not matter. Only the encoder is kept, generating the 2 dimensional representations of the data it was trained on.

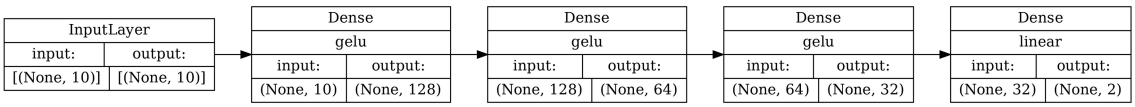

Figure 2: Encoder part of the autoencoder network implemented in TensorFlow 2.x framework. Input dimension is 10, latent dimension is 2.

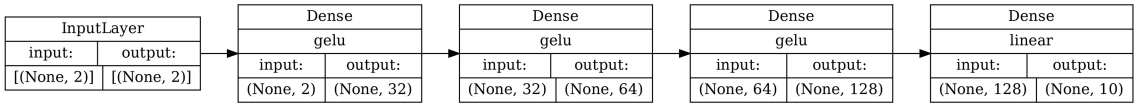

Figure 3: Decoder part of the autoencoder network implemented in TensorFlow 2.x framework. Latent dimension is 2, output dimension is 10.

3.4.2 Autoencoder training

The autoencoder is trained on a simple task. “Generate the same data on the output as received on the input, but first pass them through a ‘bottle neck’ inside the network”. Thus imposing on the model to learn low dimensional encoding of the input in the latent space. Loss used during training is usually called a *reconstruction loss*, but in reality it could be the plain old *MSE*

(Mean Squared Error) *[35]* loss, if the model output are *logits* (raw, unprocessed activations of the last layer).

3.4.3 Autoencoder and Principal Component Analysis similarities

There is a similarity in the resulting projection between the *PCA* and the encoder part of an Autoencoder, if constructed in the most simple way, both encoder and decoder having only 1 hidden (fully connected layer) without any activation. After training to convergence, such encoder generates the same subspace as the *k*-largest components identified by PCA, where *k* represents the size of the latent dimension in the autoencoder. However, the resulting projection may (and most probably will) differ when additional layers or non-linear activation functions are introduced. *[36]*

3.4.4 Variational autoencoder

Variational autoencoder could be viewed as a regularized version of the plain autoencoder. For the plain autoencoder the loss is calculated only as a difference between the input and the 'reconstructed'. For Variational Autoencoder, there is put an additional constrain on the distributions generating the latent space. The combined loss is usually described as (1) or, in order to formulate the training as a minimization task, as (2) and it is a called an *Evidence Lower bound.*

$$L_{ELBO}(\theta, \phi; x) = \log P_\theta(x) - D_{KL}(Q_\phi(x|z) || P_\theta(z|x)) \quad (1)$$

$$-L_{ELBO}(\theta, \phi; x) = \log P_\theta(x) - D_{KL}(Q_\phi(x|z) || P_\theta(z|x)) \quad (2)$$

VAE is trained by minimizing the $-L_{ELBO}$. *Distri*bution *Q* is parametrized as Normal distribution $N(z| \mu, \sigma^2)$ with encoder generating the mean μ and variance σ^2 for given input **x**. The expected value of $Q_{fi}(z|x)$ is estimated using single sample, prior is P(z) ~ N(0, I) and loss has 2 parts:

→ Reconstruction loss – Start with input **x**, pass through **Q**, sample **z**, pass it though **P**, should output **x.**

→ latent loss – over all **x**, the distribution $Q_{fi}(z|x)$, should be as close as possible to the distribution $P(z) = N(0, I)$, which is independent on x. Backpropagation is not possible due to sampling. In order to backpropagate through $\boldsymbol{z} \sim Q(z|x)$, "*re-parametrization trick*" must be used. *[23]*

The theory for the *VAE* was described into more detail, because unlike the majority of the algorithms used out-of-the-box, *VAE (*and *AE)* had to be implemented from scratch.

3.4.5 Variational Autoencoder architecture

Figure 4 and *Figure 5* depict the actual internal architecture of Variational Autoencoder, as later implemented in this work. The interesting part of the network, are arguably the two output layers of the encoder (*Figure 4*), which generate two separate outputs representing the mean and the variance (here both two-dimensional vectors) used to parametrize the Normal distribution. This parametrized Normal distribution is then used for sampling encoder output, which represents the network input in the latent space. This is done in order to 'cover' the whole latent space. Each model input generates not a point but a probability distribution, rather than projecting every the input every time into one point as the Autoencoder does.

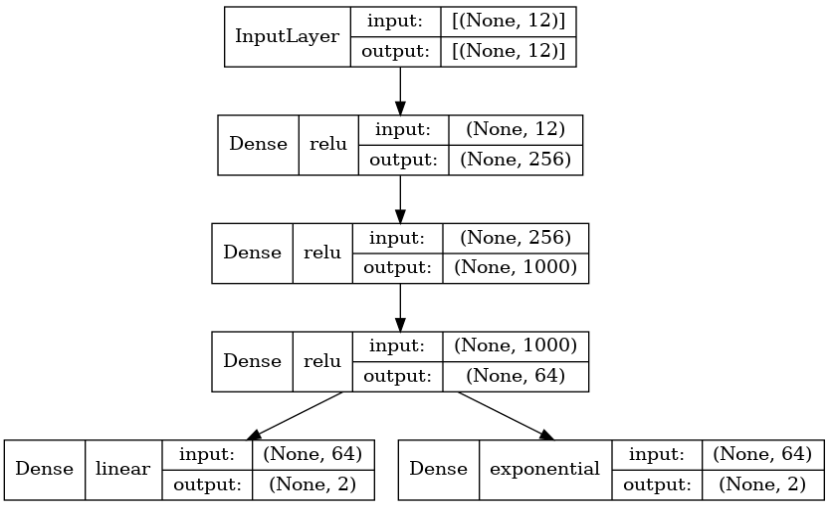

Figure 4: Encoder part of Variational Autoencoder. This is the actual architecture later implemented in this work. Input layer expects batch of vectors of size (12,). The last two separate layers generate the mean and the variance used to parameters the a Normal distribution used for sampling the data representations, but in the latent space, which has here 2 dimensions.

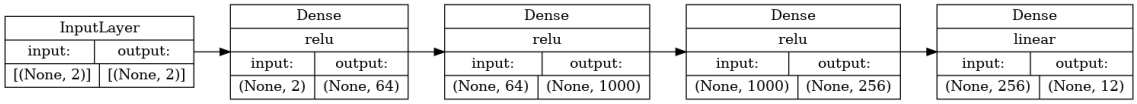

Figure 5: Decoder part of Variational Autoencoder. Input to this model is a 2-dimensional vector in latent space, sampled from the Normal distribution, parametrized by the encoder output. Three hidden fully-connected layers follow. Model outputs vector in the same shape that expects the encoder on the input, here (12,).

3.5 Metrics

Almost all algorithms discussed in this chapter need some way of measuring the pairwise distances between the data. This usually defaults to some kind of *L1* or *L2 Minkowski norm* (Manhattan or Euclidean distance respectively). Large number of algorithms implementations directly offer a plethora of metrics to be used out-of-the-box, for example *UMAP*[2] python module offers following groups of metrics:

- 'minkowski-style' metrics
- angular and correlation metrics
- normalized spatial metric
- metrics for binary data

However metrics that can compute the pairwise distances directly do exist. One of such metrics would be the *Gower's distance.*

3.6 Unsupervised clustering

This section describes unsupervised learning algorithms used in this thesis for creating the patients digital phenotypes. From number of clustering algorithms available, as discussed in *[25]*, following 4 were chosen, namely: *hdbscan [37]*, *optics [38]*, *agglomerative hierarchical clustering [39]* and *k-means [40]*. *K-means* was mainly chosen, in order to have in the clustering arsenal some 'base-line' well known algorithm. That doesn't mean that for some problems this method cannot perform sufficiently. Algorithms are presented from a practical point of view, this section doesn't aim to provide exhaustive overview of each used method, such as have been done in *[41]* and *[42]*. Most algorithms work with some kind of *intra-cluster* (inside the same cluster), and *inter-cluster* (between distinct clusters) similarities. Metric must be provided to the algorithms in order to compute the cluster distances. This presents very similar issues, already discussed in previous section dedicated to dimensionality reduction algorithms. *K-means* is a well known method. It's major hyper-parameter is the number of clusters, that need to be set prior to starting the algorithm. This task can be tackled by using various 'elbow methods' or using the *gap statistics* (discussed in section *Clustering evaluation metrics*), a more advanced method that automates the process of choosing the number of clusters. Agglomerative clustering, as well being an older concept, proves to be a powerfull technique. It recursively merges closest pairs of clusters of the sample data, provided some chosen *linkage function [43]*. *Hdbscan* and *optics* are rather novel approaches. Hdbscan is a complex algorithm, but the major parameter of it's 'interface' is arguably the *minimal cluster size*. *Optics's* major parameters set how the neighborhood of points is constructed and what points will be chosen as the *core* points.

3.7 Clustering evaluation metrics

This section starts by describing used performance evaluation metrics used to asses results obtained by the clustering algorithms described in the previous section. Then, another algorithm called *gap statistics[44]* is described, a method used to choose the optimal number of clusters.

There are multiple metrics applicable for evaluation of the clustering performance, only a few can be used for clusters without knowing the ground truth labels. These methods usually work with some notion of how well separated the clusters are. Following methods require only the data and according predicted cluster labels on the input: *Davies-Bouldin Index [45]*, *Calinski-Harabasz Index [46] (*also known as *Variance ratio criterion),* and *Silhouette coefficient [47]*. Again, all these methods need to be provided with some distance function used for calculating the distances between the data and clusters, and again this usually defaults to Euclidean distance. There are also other metrics such as *mutual information* or *rand index*, but these methods require both predicted and ground truth cluster labels on the input, so they cannot be used for our problem. Short summary is presented in *Table 5.*

Table 5: Clustering evaluation metrics, applicable without knowing the ground truth labels, with value ranges and their meanings.

Metric name	Range of the score	The higher the score, the more dense and well separated the clusters are.
Davies-Bouldin Index	<0, inf>	No, the lower the better
Calinski-Harabasz Index	<0, inf>	Yes
Silhouette coefficient	<-1, 1>	Yes

3.7.1 Gap statistics

This method offers a convenient way for choosing an optimal number of clusters for given clustering algorithms, that work in some sense with cluster centroids. Thus, it cannot be directly used for *hdbscan* or *agglomerative hierarchical clustering.* The authors in the *original paper* however state, that it can be used with any clustering algorithm. So it is possible, that this could be only an implementation-wise issue, of the used third-party algorithm implementation. Nonetheless, only 1 credible implementation in python was found, the other - more famous one - is implemented in R. It is usually used for k-means algorithms family. As stated in *[44]* this method aimed on formalizing some of the heuristics used for choosing an optimal number of clusters. Here optimal is defined as minimizing the *within-cluster dispersion.*

3.8 Encoding categorical variables

Neural networks, and large portion of methods and algorithms used in this thesis, require numeric input values (if non-exotic metric, such as Euclidean distance is used). There isn't a single definitive solution for handling categorical data. The encoding should be done cautiously, with regard to the specific method and chosen metric. Also different encodings could be more, or less, suitable for the categorical variables. It usually depends whether it is nominal or ordinal categorical variable.

First, encoding methods overview is given, followed up by presenting possibly, to our best knowledge, a novel way of learning *deep embeddings* with multi-layer perceptron network, formulating the task as a multi-label classification *[48]*. Many encoding methods exist, quick refreshment of the most commonly used ones follows.

3.8.1 One-hot encoding

One-hot encoding de-factor splits the categorical variable into as many binary variables, as there are categories, creating (usually) a sparse vector. For example nominal variable '*state of health*' with values {*cancer, healthy, diabetes*} would be one-hot encoded into *cancer*$\rightarrow$*[1, 0, 0]*, *healthy*$\rightarrow$*[0, 1, 0]*, *diabetes*$\rightarrow$ *[0, 0, 1]*. Creating 3 new dimension, each representing yes/no answer to the question "Is your state of health <*category*>". Used mainly for categorical-nominal variables, it tends to inflate the dimensionality quickly. The number of newly created dimensions is equal to the number of the categories. If used for encoding ordinal variables, the implicit ranking is lost.

3.8.2 Label encoding

This term usually refers to the method of simply assigning consecutive integers ('labels') to the categories. For example, variable *town_size={small, medium, large}*, is label encoded as follows: medium$\rightarrow$1, small$\rightarrow$2, large$\rightarrow$3. This - valid - labeling was chosen on purpose to demonstrate, that if the labels are afterwards interpreted as real numbers, the 'large town' would be closer to 'small town' than to the 'medium town', under euclidean distance. Usually, it makes sense to use this method for ordinal variables, because for nominal variables, it puts unrelated values falsely close together. However, it presents no problem to use this encoding for nominal variables if they are directly fed into a neural network. Notion of a distance or similarity is usually captured by the task the model is trained on.

3.8.3 Entity Embeddings

This approach leverages the power of neural networks to *train* the embeddings, provided the input data, some target and suitably formulated loss *[49]*. Resulting embedding is a dense vector of fixed size with real values. The architectures may vary from simple multilayer perceptrons to very deep and complex networks, but they usually share what is referred to as an '*Embedding layer'*. The input to the Embedding layer is interpreted as a key to a dictionary, with values being the actual embeddings, which is then passed on the layer's output.

3.8.4 Entity Embeddings for subsets

Learning entity embeddings are generally used to obtain a vector representation of each word from the dictionary, where word represent a label encoded category. Problem encountered in this work is more specific though. The issue is, that there is an additional information hidden in the nominal variables values. For example, the values of variable *state of health {2, 5, 6}* and *{2, 3, 6}* are obviously more related than the values *{1}* and *{2, 6}*. To capture such relations, appropriate formulation of the problem has to be used. The model input will be still following an 'usual procedure'. First encoding the categories with *Label encoding*, using those values as an input to the neural network model. To capture then the similarities between categories, that is the subsets intersection, training the of the model formulated as a multi-label classification task is proposed, where the target is a binary vector of size *[1, number of variable's options]*. This formulation dictates that *binary cross-entropy loss* (also known as the *log loss*) should be used, defined as:

$$-\frac{1}{N}\sum_{i=1}^{N} y_i . \log\left(p\left(\vec{y}_i\right)\right) + \left(1-\vec{y}_i\right) . \log\left(1-p\left(\vec{y}_i\right)\right) \tag{3}$$

where N is the length of the target vector, $\mathbf{y}$ is the target vector, $p(\mathbf{y}_i)$ is the probability of label i being 1 and $p(1-\mathbf{y}_i)$ being 0.

Beware, the 'label' in the word 'multi-label', doesn't have the same meaning as in the word 'label encoding'. It represents the cardinality of the set *{1, …, n}* (*n* being the number of options patient chooses from in some questionnaire), not the number of classes – all subsets of the set *{1, …, n}*.

For example, the multi-label target representation for the value *{2, 3, 6}* of *state of health variable* (with 7 options) would be *[0, 1, 1, 0, 0, 1, 0]*. We could go even further and weight each of the labels (options) by number of times it is present in the dataset, because it is presumable the number of observed options would be unbalanced.

3.8.5 Entity Embeddings for subsets implementation details

This paragraph discusses the technical aspect of constructing the neural network model used for learning the nominal variables embeddings. The model was implemented in the *TensorFlow 2.x* using its *Sequential Model api*, which is sufficient for this task (*TF* offers more advanced *api* called *'functional'*, which used for implementing *VAE*). As stated previously, the general idea of this approach is to learn, on some suitably defined task, the weights of the Embedding layer. Weights of this layer then directly represent the embeddings of the categorical variable values. The size of the Embedding layer – and the resulting variable embeddings – is *[size(vocabulary), embedding dimension]*. Vocabulary size is equal to the number of categories of the nominal variable, in our case all subsets (present in dataset) of set *{1, …, n}*. Embedding dimension is a hyper-parameter. Higher the embedding dimension, the more capacity the model has to encode nuances in data it's trained on. However, this presents a trade-off between how well the categories can be represented by the embeddings and increasing dimensionality on the other hand. There are many 'rule of thumbs' found in literature for choosing the embedding dimensionality. Usually, for *nlp* tasks, this could in order of hundreds or even thousand of dimensions. However, for variables of relatively small vocabulary size, as found in this thesis, only a few embedding dimensions seems to be sufficient for the model to converge during the training. Optimizer used is *AdamW.* As loss, the *BinaryCrossentropy(from_logits=True)* is used, with the default hyper-parameters settings. Raw network outputs without any activations (logits) are passed into *n* separate sigmoid activations, giving *n* independent probabilities, 1 for each label (label meaning the option, not category). Unlike *softmax*, which would yield 1 probability distribution for all the *n* labels, which would then dictate to use 'c*ategorical cross-entropy loss'* instead.

Figure 6 shows the architecture of implemented neural network used to learn the embeddings as they were just described. The embeddings can be directly extracted as the weights of the '*Embedding*' layer.

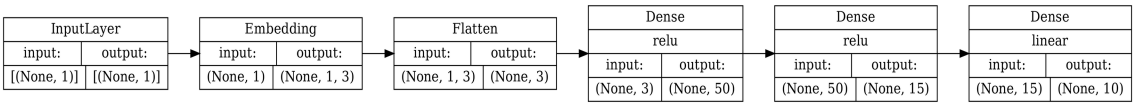

Figure 6: Neural network architecture used for learning entity embeddings. The model outputs a vector of logits of size [1, number_of_options] which is point-wise passed to a sigmoid activation and then used to compute the binary cross-entropy loss with the multi-label target. Sizes of input and output tensors are (batch_size, dim_1, …, dim_k).

3.8.6 No encoding

Keeping a mixed dataset is also a relevant approach. Instead of unifying the feature types, distance metrics allowing mixed dataset can be used, for example the *Gower's* distance. Some clustering algorithms can directly work with mixed datasets, for example the decision trees or the decision forests.

3.9 Feature scaling

3.9.1 Scaling numerical variables

As stated previously, majority of the algorithms calculate pair-wise distances. If the dimensions (features) were presented to the algorithms while being on different scales, the results would be naturally skewed by those scale-differences. Also many methods assume, the data distribution ate least resembles the Normal distribution. Therefore, transforming data onto similar scales is necessary step prior to performing the experiments. The two major and very simple methods are usually used. *Normalization*, also known as *min-max scaling* and *Standardization* also known as *z-score normalization.* In general, Normalization should be used when the data doesn't have Gaussian distribution. Standardization should be used on data having Gaussian distribution.

3.9.2 Scaling ordinal categorical data

Ranked data can be scaled as well, assuming that the differences between ranks are the same. Strategy, used further on in this work, is to scale ordinal variables to the range from 0 to 1, with equal step. For example, ordinal variable with ranked categories *1, 2, 3, 4, 5* would be scaled to *0, 0.25, 0.5, 0.75, 1*.

3.9.3 Scaling embeddings

The 'deep embeddings' retrieved directly from the neural network, can be of arbitrary scale. There is no objective forcing the model to do so. It could be appropriate to scale these embedding values as well.

Chapter 4

Creating the dataset

In order to perform the experiments described in the previous chapter a dataset must be created. This chapter describes all the steps in the creation of real chronic nicotine users dataset. Firstly, it is described how the unprocessed data are retrieved. Then are presented the patients features chosen to be used for the experiments. Later the preprocessing steps are presented.

4.1 Querying the database

Patients data are retrieved on each run of the application[2] from the PostgreSQL database server. This database server contains multiple tables, but the ones queried for the chosen patients features are *users_user* and *payments_payment*.

SQL query itself is written in PostgreSQL dialect (script *retrieve_user_data.sql).* The query is kept simple, most of the preprocessing is done later using python. Majority of the relevant features are stored as *json* objects inside one single column '*data'* in the table '*users_table'*. PostgreSQL offers a convenient way to work with *json* format objects stored directly in the database, so the parsing of the *json* objects is done directly in the query. The *json* feature values are parsed as *text* type, conversions to the correct data-types are handled further downstream in the pipeline. Retrieved data are *pickled* (python term for serialization), adding a timestamp. If pickled dataset with timestamp of the current day is found on the application start, the pickled dataset is used to speed things up, because the query takes about 30 seconds to complete. Otherwise the old dataset is discarded and new up-to-date dataset is retrieved from the database.

2

2 Implemenation uses the *psycopg2* python module for interacting with the database server.

Table 6 presents the 25 retrieved variables, their naming in the application and names used further on for convenience.

Table 6: Retrieved features from the smoking cessation app database, with their naming in the database and the naming used further on in this work.

Feature name in the database	Feature name used further on
region	region
date_joined	date_joined
state	app_purchased
0Sex	sex
AnaData_userAge	age
PHASE	therapy_phase
stLM	lapse_count
NoCig0	cigarettes_per_day_before_therapy
NoCig1	cigarettes_per_day_now
BujonOrEE	bujon
AnaData_userIncome	income
AnaData_smokingSince	smoking_since_regularly
AnaData_products	tried_nicotine_product
AnaData_usage	product_using_at_least_once_a_week
AnaData_reason	reason_for_quitting_smoking
AnaData_attempts	quitting_attempts_count
AnaData_townSize	town_size
AnaData_userJob	employment_type
AnaData_userEducation	educational_background
AnaData_userHealth	state_of_health
AnaData_userCovid	covid_suffered
AnaData_userMedicaments	taking_medication_regularly
AnaData_methods	last_withdrawal_method
AnaData_lastDuration	last_non_smoking_period_duration
ReaQui	reasons_for_quitting_smoking

4.2 Features description

This section describes in more detail, what retrieved sociodemographic and other patients features stands for. It is logically divided by the variable type into 4 groups: binary – *Table 7*, ordinal – *Table 8*, nominal – *Table 9* and numerical – *Table 10*. The data were gathered from January 2018 until May 2023. The variables ranges and valid values are presented after

preprocessing and visualized during the experiments. All the values and types are stored in the script *features_metadata.py* .

Table 7: Binary features and their description. 0 represents No, 1 represents Yes.

Feature name	Feature description
region	The country user resides in. '*CZ*' for Czech Republic, '*RoW*' – rest of the world, but mainly Norway.
sex	Patient's sex. 0 represents a woman and 1 represents a man.
covid suffered	Patient suffered covid19.
taking_medication_regularly	Patient takes some kind of medication on the regular basis.
app_purchased	This variable indicates whether the patients purchased the application, enabling them to complete the entire procedure. The trial version concludes after the *EE* phase, so this variable is utilized to further split the dataset into two groups: patients who are able to complete the therapy and those who are not.

Table 8: Ordinal features and their description.

Feature name	Feature description
therapy_phase	The furthest achieved application therapy phase. Patients with trial, non-paid application version, cannot pass further than the *EE* phase. This implies that for the trial dataset some metrics, for example whether patient finished therapy, cannot be measured.
town_size	Size of the town patient resides in.
educational_background	Highest achieved patient's education.
last_non_smoking_period_duration	Categorized from shortest to longest, last patient's cessation period.

Table 9: Nominal features and their description. The values of those variables are responses from a questionnaire, where patient can choose multiple options at once.

Feature name	Feature description
tried_nicotine_product	Nicotine products patient tried so far.
product_using_at_least_once_a_week	Nicotine products patient uses at least once a week.
reason_for_quitting_smoking	Patients reasons for quitting smoking.
last_withdrawal_method	Last tried nicotine addiction withdrawal method.
employment_type	Patients current employment.
state_of_health	Patients state of health.

Table 10: Numeric features and their description.

Feature name	Feature description
age	Patient's age in years.
lapse_count	Number of relapses - times patient used a nicotine product after declaring nicotine abuse cessation. Only measurable on patients with paid *app* version.
cigarettes_per_day_now	Number of cigarettes per day patient smokes now, during the ongoing the therapy.
cigarettes_per_day_before	Number of cigarettes per day patient used to smoke before starting the therapy.
smoking_since_regularly	The age in years since the patient smokes regularly.
quitting_attempts_count	Number of times patient tried to quit nicotine abuse .

Variable *date_joined* was mainly used for comparing the feature values in the dataset before and after some major changes in application. Retrieved mainly for debugging and testing reasons. The variable *reasons_for_quitting_smoking* was only retrieved and inspected, but isn't further used.

4.3 Dataset preprocessing pipeline

Following section describes how the retrieved data is processed into needed for later experiments. Variable values retrieved from the Adiquit database are in a rather raw state. They are obtained directly as an input by the patients, Adiquit application users, with small to none validation. To such state contribute also the technical aspect of handling the data internally in the application (presumably passing them through a various data structures or performing format conversion). Most of the variables contain either invalid characters, values outside valid range, NaN values or unordered lists representing the subsets for multi-option nominal features. Such impurities differ for each of the variables. For further analysis each variable had to be taken care of individually and it was done in most conservative manner, not to discard any possibly useful information in the early stage of the analysis. For example: if the user inputs as the number of smoked cigarettes per single day some astronomical number, let's say a thousand, it could mean two things. The first one could mean, that it is a typo or the user didn't care to input a valid number. The second explanation could be, that the patient actually did express a sentiment, feeling like smoking a thousand cigarettes a day. That would be just a one example to illustrate that blind range-clipping could lead to information loss, so this phase is done in the most circumspect manner.

4.4 Pipeline steps

The preprocessing pipeline[3] performs the following steps:

1. Drop unused features.
2. Drop rows with null (NaN) values.
3. Validate all features.
4. Create new features.
5. Encode nominal variables, choosing on of the following approaches:
    - one-hot encoding, splitting the variables, possibly making them less granular
    - learning the deep embeddings
6. Scale all the variables (using normalization or standardization).

1. For various reasons (e.g. debugging purposes) more features are retrieved from the SQL database server, then they are finally used for during the experimental part of this work[4].

2. Removing missing values for the predefined set of features. There is a trade-off between the number of features and number of patients that will be kept after this step, this is a opinionated decision. Different approach would be trying to interpolate the mising values for given features. But this seemed for this dataset rather impractical, and also the size of the resulting dataset is still being reasonably large.

3. Values validation. This step ensures that only valid values are present in the final dataset. For each feature, it handles data types conversions, valid ranges checks for the numerical data, valid values from the predefined set of possible values for the categorical data, etc.

4. New features are created: *finished_therapy* and *smoking_category. Finished therapy* is created as binary variable and is later used as the target variable for determining whether the therapy was successfully or not. It is equal to 1, only if patients *therapy_phase* (see *Table 8*) is equal to *FIN (see Table 1)* and the lapse count is 0. S*moking_category*, ordinary variable, represents number of smoked cigarettes per day but less granularized, having only 3 values l*ight, medium* and *heavy* smokers. Motivation behind adding this variable is that in literature smokers are usually divided into following into those 3 categories.

[3,4]

3 Implemented by the DataCleaner python class, scrip data_cleaner.py .

4 Features to be dropped could be passed to DataCleaner constructor, otherwise defaults are used.

4.5 Nominal variables

The variables which represent patient choosing multiple options from a questionnaire have usually high cardinality. For example nominal categorical variable '*reasons for quitting smoking*' have 7 options, which renders $2^7=128$ possible values. Large portion of those possible values however would not be present in the dataset or their value count would be very small. The best solution to this problem seems using the Entity Embeddings as described in Methods. *Figure 7* shows the learned 3 dimensional learned Entity Embeddings. It can be seen, how the similar values – sets with some intersection, are grouped together. On the other hand, disjunctive sets are further away.

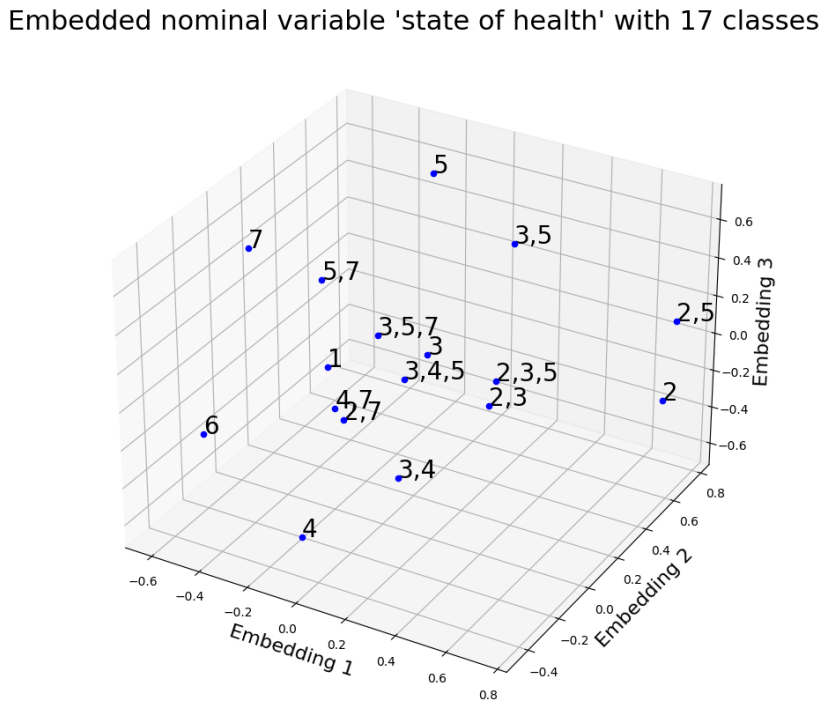

Figure 7: Visualization of 3-dimensional embeddings Entity Embeddings learned by neural network of the nominal variable 'state of health', whose values are subsets of set={1, 2, 3, 4, 5, 6, 7}.

4.6 Full-version, trial-version and combined datasets

After performing all the described preprocessing steps, dataset is further split into two parts, by the *app_purchased (see Table 8)* into *trial-version dataset* and *full-version dataset*. If the both datasets are combined, it is referred to as *combined dataset* further on.

Chapter 5

Results

This chapter reports the all the analysis steps, all the performed experiments and their results. It starts by the exploratory analysis. Firstly, by univariate analysis, introducing each one of the variables separately. Bivariate analysis follows, providing some insight into relations between the variables. First for the variables of the same type, than also for the mixed variable types. Third step is the multivariate analysis which tries to discover even more patterns in the patients dataset. Then, dataset is projected to 2 dimensional space using the according methods discussed in Chapter 3. Gathering all the information along the way, what follows is the actual clustering using methods again discussed in the third chapter. Subsequently the clustering performance evaluation is performed. Finally the digital phenotypes are calculated, either by taking mean values or as an intersection of the subsets for the nominal variables.

Most of the results are for the full dataset, that means both patients with trial and full app version. Whenever used the full-version dataset, it is explicitly stated before presenting the results.

5.1 Univariate analysis

This section starts by inspecting numerical, binary, ordinal and finally nominal variables. For each variable a short discussion is given.

5.1.1 Numeric variables

Following results describes the distributions and moments of numerical variables, namely: age in years, age when patient started smoking in years, number of smoked cigarettes per day and number of cessation attempts prior to starting the therapy, for combined dataset. The lapse count has meaning only for the full-version dataset.

The average patient is ~31 years old, but the variance in age among users is quite high, standard deviation being almost 10 years. Third and fourth moments values are not surprising, because the age is more limited from the left. Depicted on *Figure 8.*

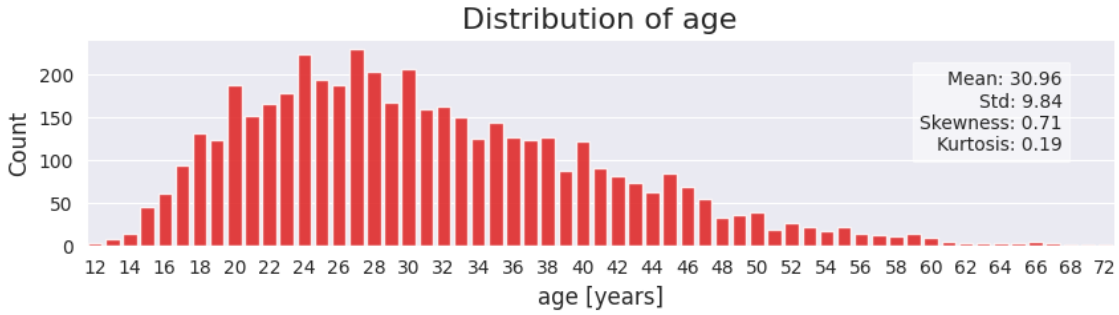

Figure 8: Age distribution, for combined dataset. Average age of patients is ~31 years.

It seems that majority of the patients smoke 1 pack of a cigarettes a day, or maybe the peak is so present at 20, because the question is too granular and it is hard for patient to say whether the precise number of smoked cigarettes per day is 17, or 22. So similar peaks, representing the multiples of 1 pack of cigarettes, emerge. Half a pack – 10, ¾ of a pack – 15, 1 and ½ a pack at 30 and two cigarette packs at 40. This variable could be further transformed into and categorical ordinal, using less granular scale. Depicted on *Figure 9.*

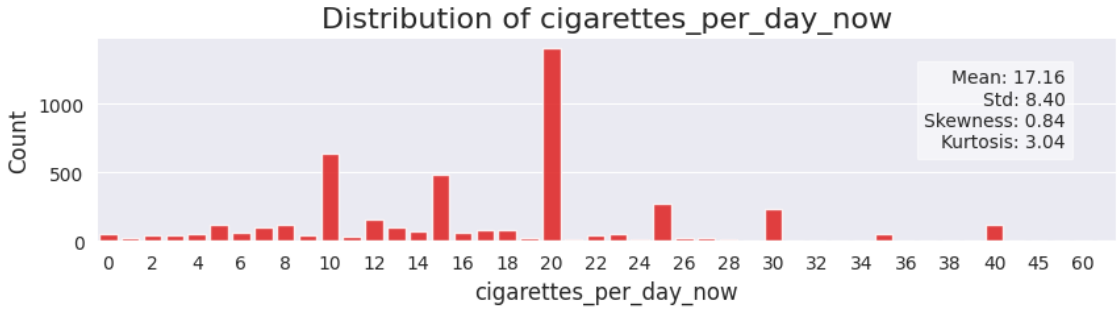

Figure 9: Distribution of smoked cigarettes per day, for combined dataset. Peaks representing multiples of 20 cigarettes are visible (1 pack of cigarettes)

The distribution of cessation attempts has two peaks. The first obvious group are the users that tried breaking the habit, but were unsuccessful up to 5 tries. Than are the patients that tried unsuccessfully many times, the values above 10 attempts were set to 10. *Figure 10:*

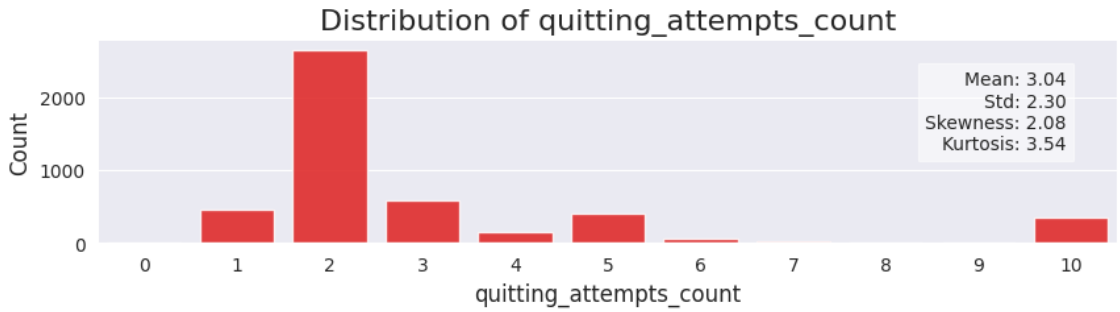

Figure 10: Distribution of number of times patients tried to quit nicotine abuse prior to the therapy. Two subpopulations can be observed. Those with lower, approximately 3, number of previous attempts and those who were unsuccessful many times, 10 and more.

Majority of patients starts smoking around 15 to 17 years of age. Depicted on *Figure 11:*

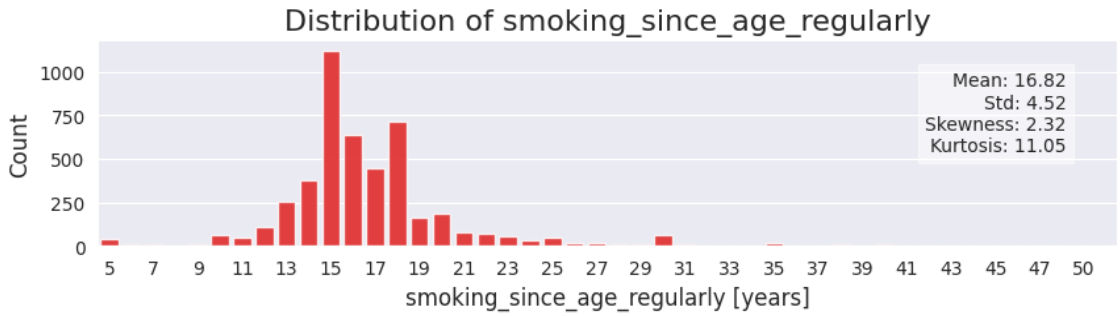

Figure 11: Distribution of age when the patients started smoking.

The distribution of the patients relapses resembles Poisson probability distribution. Majority of patients have no relapse. This feature is measurable only on the full-version dataset. *Figure 12:*

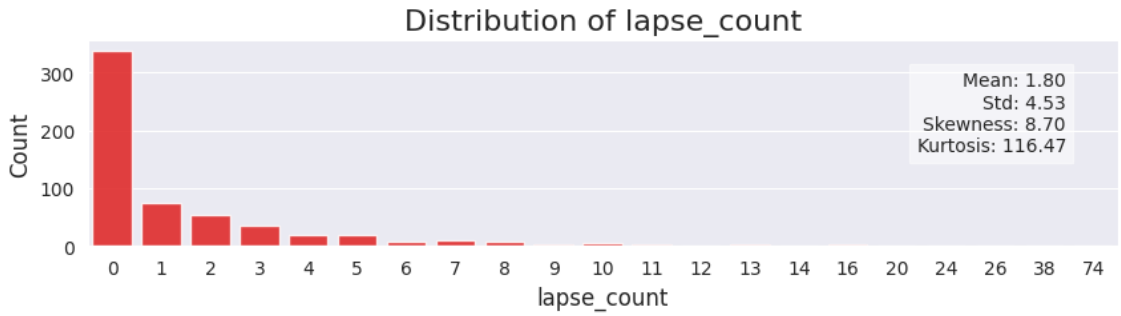

Figure 12: Distribution of relapses, measurable only on the full-version dataset. Majority of patients did not relapse.

5.1.2 Binary variables distributions

The sex is almost evenly distributed, majority of ~88% of patients are from the Czech republic, minority of ~23% of patients suffered covid19 and ~73% of patients take some form of medications regularly. *Figure 13:*

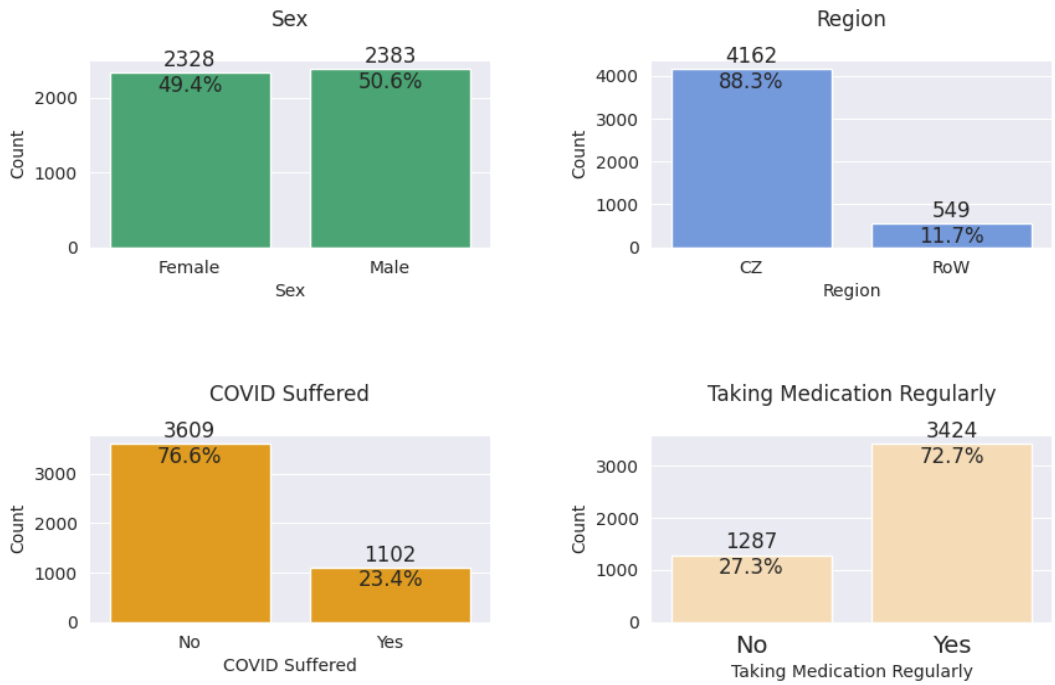

Figure 13: Binary variables: sex, region, if suffered covid19, if taking medication regularly. CZ stands for Czech republic, RoW for rest of the world.

5.1.3 Ordinal variables

Following plots shows the distribution of values of each of the categorical ordinal variable, that represent the patients educational background, furthest achieved therapy phase, categorized length of last non-smoking period and the size of the town patient resides in.

The the patients populations roughly splits into two groups by the town size, those having finished only the high-school and those that have some university diploma, with sizes 2/3 and 1/3 respectively. Depicted on *Figure 14.*

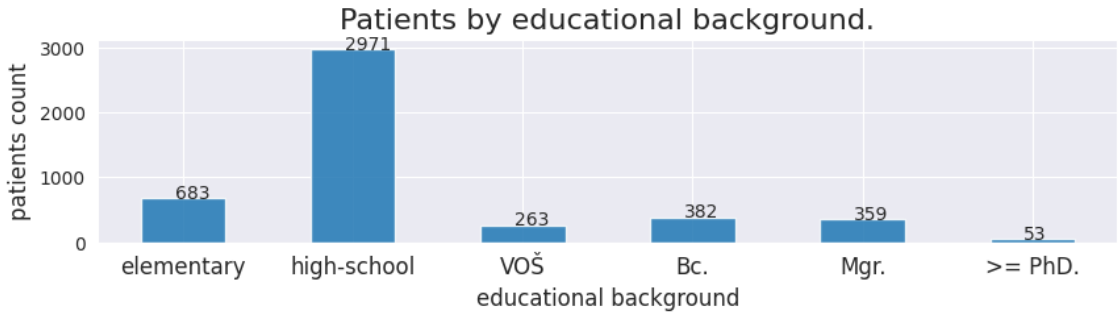

Figure 14: Highest achieved education level. Majority of the patients have only finished the high-school. Second most present group are patients with master diploma. Patients population roughly splits into two subgroups, those with high-school (~2/3) and those with some kind of university education (~1/3)

Majority of the patients managed to stop smoking only for 1 week. Second most populous group are the patients who managed not to smoke up to 1 month. Full distribution of variable *last non-smoking period duration* on *Figure 15.*

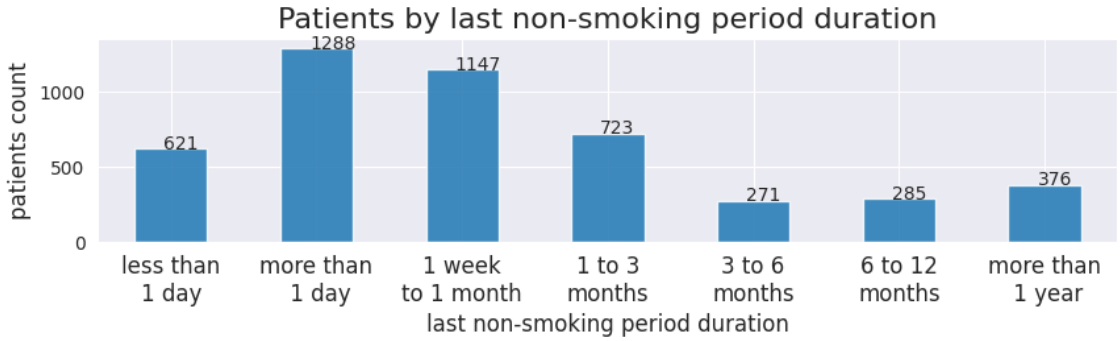

Figure 15: Distribution of user last non-smoking period. Majority of chronic smokers manage to cease smoking only for shorter period of time, up to 1 or 3 months, before they relapses. Minority of patients manages to stay nicotine-free for longer period of time.

The distribution of sizes of towns the patients reside in is quite uniform, however patients from medium-size to small-size towns are more prevalent, *Figure 16:*

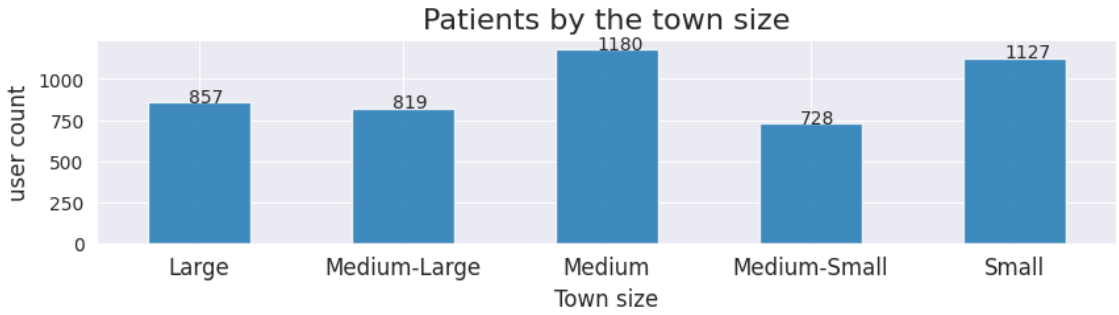

Figure 16: : The distribution of patients regarding the size of the town they reside in. The distribution is quite uniform. The 'Medium-Large' and 'Medium-Small' might be chosen less often, because the question is too granular, majority of user choosing between 'Large', 'Medium' and 'Small'.

The therapy phase analyzed **only** for the full-version dataset. This feature is not possible to measure on the trial-version dataset. Most of the patients managed to use the application for a longer period of time. This doesn't mean, that they stopped smoking. To determine whether the therapy was successful or not, this variable must be correlated with the number of patient's relapses, see *Figure 17:*

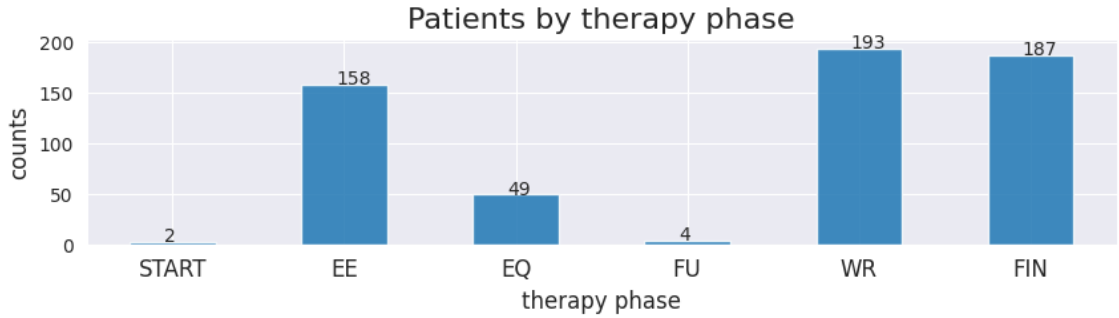

Figure 17: : Distribution of furthest achieved therapy phase in the Adiquit application. This information however needs to be further put together with other factors indicating the therapy success. By inspecting other variables such as number of lapses, it can be found out, that many users in 'WR' or 'FIN' phase, actually still smoke, because they have many lapses.

5.1.4 Nominal 'multiple-option' variables

Two strategies were implemented for this type of variables: either split the variable into multiple binary variables (one-hot encoding) or learn the deep embeddings. Preceding these steps could be merging the too granular categories. For example, the variable '*product using at least once a week'* is too much granular, containing multiple following categories: *snus, chewing tobacco, snuff, dip, nicotine pouch,* that could be described by 1 simple category 'smokeless tobacco'.

However, experiments later on use the deep embedding approach, so capturing the subtle details could be on the contrary beneficial.

Table 11 shows the number of times each option was found in the dataset for the given nominal 'multi-option' variable. The number of subsets is equal to 2 to the power of number of options. For example, variable *tried nicotine product* has by this optics $2^{12} = 4096$ categories. For such variables 'unique subsets' is the actual number of subsets was found in the inspected dataset and 'uniqueness' just represent the number of unique subsets divided by the number of all possible subsets, reported as percentage.

Table 11:Nominal features - with values (questionnaire answers) being subsets of set {1, … , n}, where n is number of options to choose from. Unique subsets represent number of found unique variable values in the dataset and uniqueness is just unique subsets divided by total number of possible subsets. Reported values are for the combined dataset.

Nominal 'multi-option' feature	Number of subsets [exponents of 2]	Unique subsets	Uniqueness [%]
tried nicotine product	12	510	10.8
product using at least once a week	12	90	1.5
reason for quitting smoking	7	262	5.6
employment type	10	66	1.4
state of health	7	30	0.6
last withdrawal method	11	82	1.7

Plots on the next pages show the counts of 'Yes' answers for each option, for each nominal variable (patient can choose multiple options at once). For example valid value *employment type* variable could be {*self-employed, part-time student*}. The bars are sorted by the number of patients that said 'Yes' to the given option. It is to be reminded, that patients could choose multiple options at a time. All graphs are for the whole dataset.

Majority of patients work a full-time job, followed by those who are self-employed or full-time students. Some of the patients have a part-time job, are on the parental leave some patients are were unemployed during the therapy. All options for employment type variable depicted on *Figure 18.*

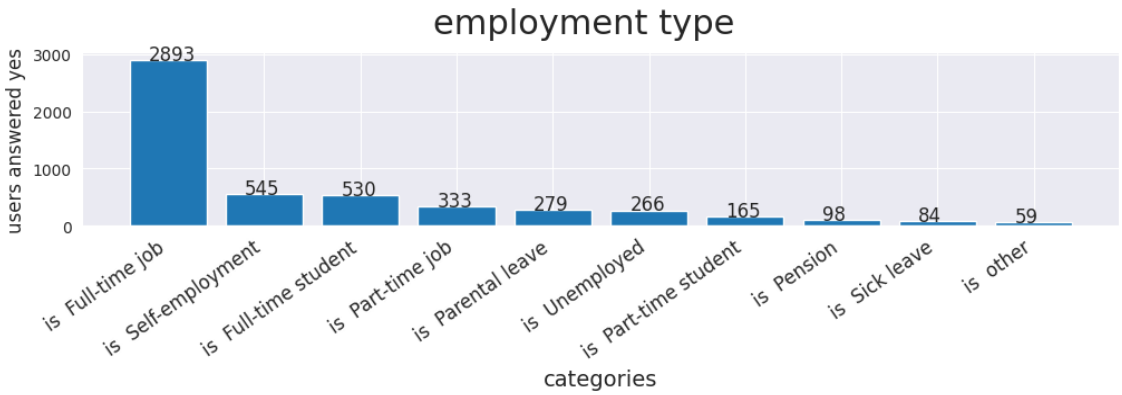

Figure 18: Number of times each option concerning the patients employment type occurred across the dataset. Most of the patients have a full-time job.

By inspecting the methods used by patients during the last cessation, can be observed that majority of patients tried to withdrawal without any help. Second most commonly used method was using some kind of nicotine substitutes. Using mobile app is on the third place. *Figure 19:*

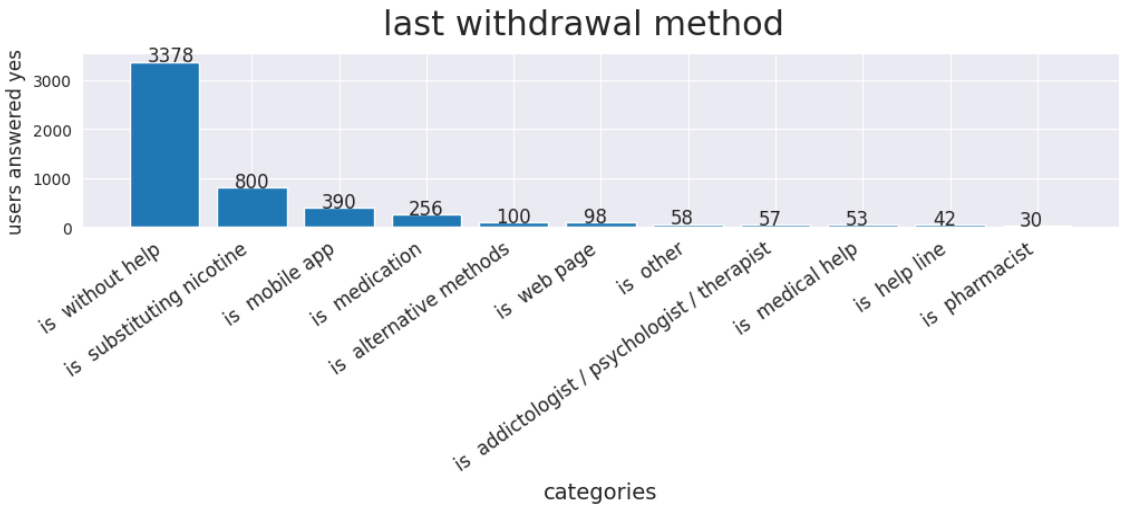

Figure 19: Methods of patients last withdrawal. Most of the users tried to quit using nicotine products the last time they tried without any help, second most common method among patients is by using nicotine substitutes and the third place takes withdrawal assisted by using a smartphone application. Some of the patients also tried medication or some alternative methods.

A Strong majority of the chronic patients in the dataset smoke cigarettes at least once a week. Second most commonly used tobacco product are the heated tobacco products followed by nicotine e-cigarettes. For the whole distribution see *Figure 20.*

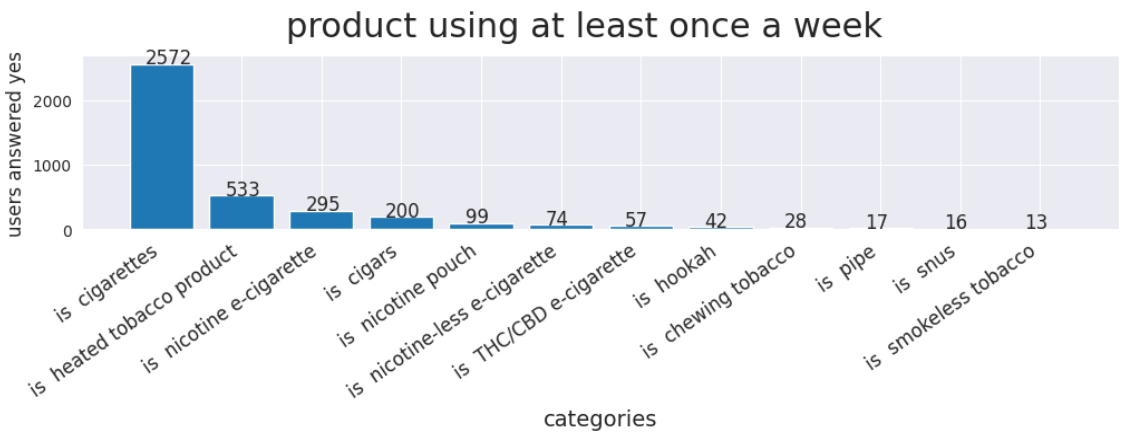

Figure 20: Products patients are using at least once a week. Nicotine products patients tend to use at least once a week. Majority of the chronic nicotine users are cigarette smokers, followed by those who use heated tobacco products, nicotine e-cigarettes and cigars.

Main reason for quitting smoking patients chose to be health. On the second place is getting addiction free, followed by financial reasons. Fourth and fifth place took family and social pressure respectively. See *Figure 21:*

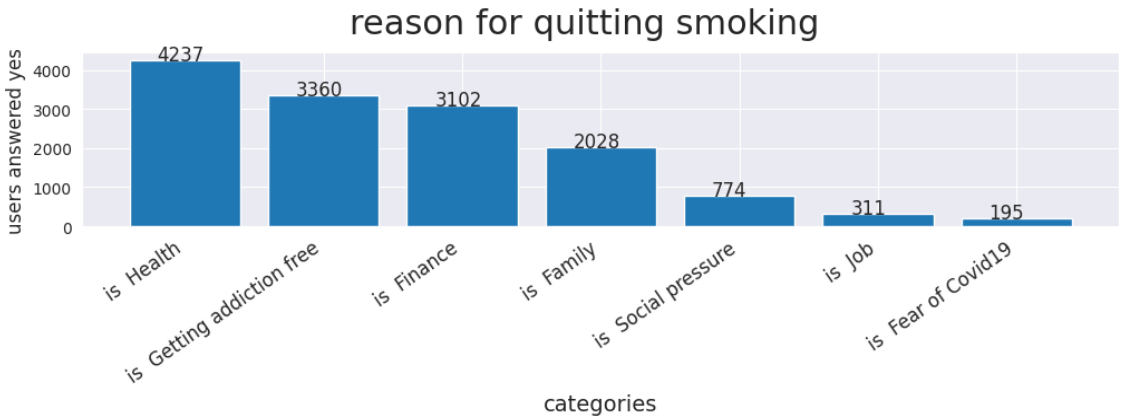

Figure 21: Reasons patients find the most important, when it comes to the motivation to quitting nicotine abuse. Number one reason 'health', second one is to get addiction free, closely followed by financial reasons. The four most common reason patients chose as 'family'.

Regarding the state of health of the patients, strong majority of patients consider themselves to be healthy. Those with some issues most commonly face some mental disorder. Second and third most common health problems are respiratory and cardiovascular diseases respectively. Full distribution depicted on *Figure 22:*

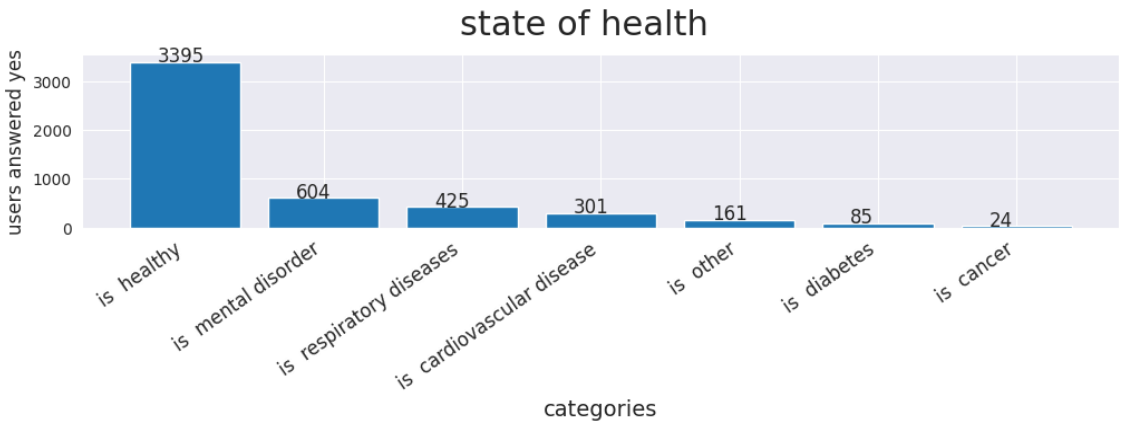

Figure 22: State of health across the cessation app users. Majority of patients consider them self to be healthy. About 13% percents of patients have some mental disorder. Less than 1/10 of patients have some respiratory or cardiovascular disease.

5.2 Bivariate analysis

Following experiments inspects the pairwise relations between variables. This is done only for the features of the same types (e.g. two numerical variables).

5.2.1 Numerical variables

It was found out that, *number of smoked cigarettes before* and *number of smoked cigarettes now* are almost identical, thus *the number of smoked cigarettes before* will be dropped. The patient's *age* and the *age at which the patient started smoking* have weak positive linear correlation. *Figure 23* shows pairwise relationships between all the numeric features. Lower triangular matrix of the grid shows the pairwise scatter plots with fitted polynomials. On the diagonal are the histograms showing number of users having the particular feature value. The upper triangular part of the grid shows pairwise *Pearson r correction coefficients (Table 2),* bigger the bubble and the number, the bigger the correlation. '*smoking since*' represents age when user started smoking, '*cessation attempts*' is the number of times user tried to get nicotine-addiction free, '*cigs now*' represents smoked cigarettes per day now, '*cigs before*' the number of cigarettes user used to smoke before starting the therapy.

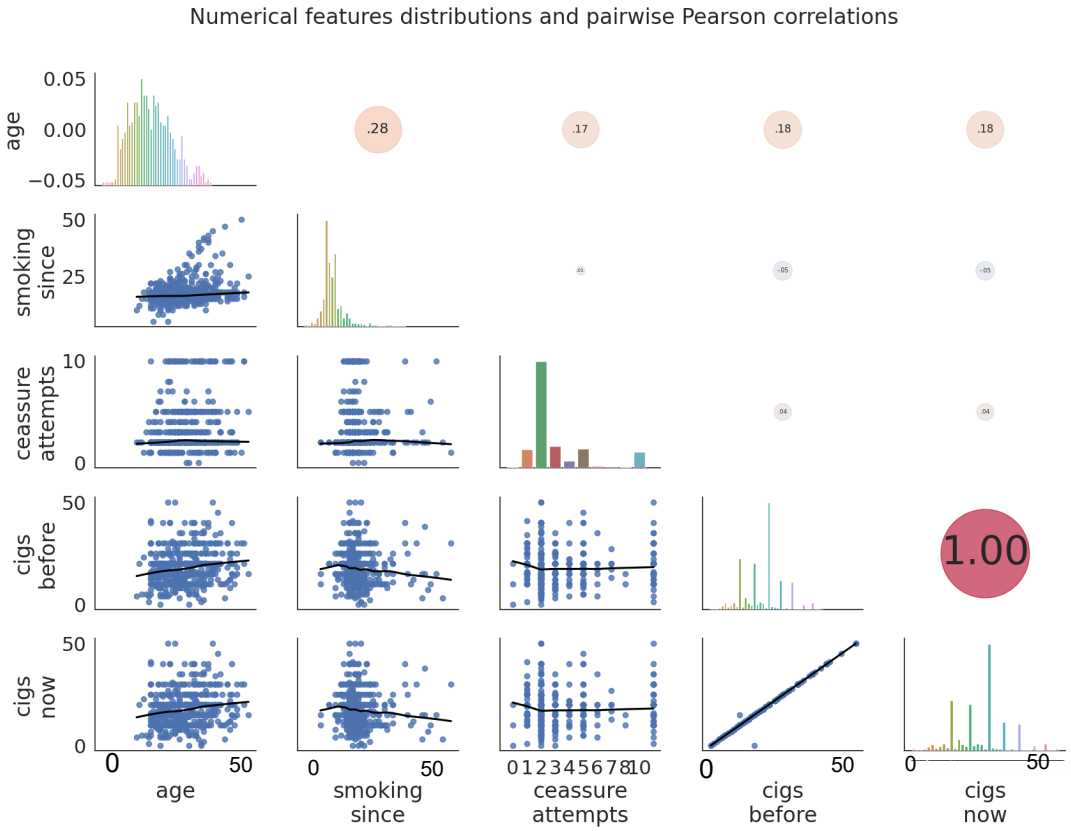

Figure 23: Pairwise scatter plots, value counts histograms and pairwise Pearson r correlation coefficients for all numerical variables from the "full version dataset". 'cigs now' represents smoked cigarettes per day now, 'cigs before' the number of cigarettes user used to smoke before starting the therapy, 'cessation attempts' is the number of times user tried to get nicotine-addiction free, 'smoking since' represents age in years when user started smoking, and age is also counted in whole years.

By looking at the correlations of numerical features and also given the small number of numerical features it can be presumed that *PCA* won't manage to reduce the dimensionality while preserving most of the variance at the same time. This assumption is confirmed by the later experiments.

Younger patients smoke on average less cigarettes. The number of smoked cigarettes per day increases until 30 y.o. where it plateaus until reaching ~45 y.o. where it increases dramatically. For average of smoked cigarettes for during the therapy per day for each patients age, see *Figure 24:*

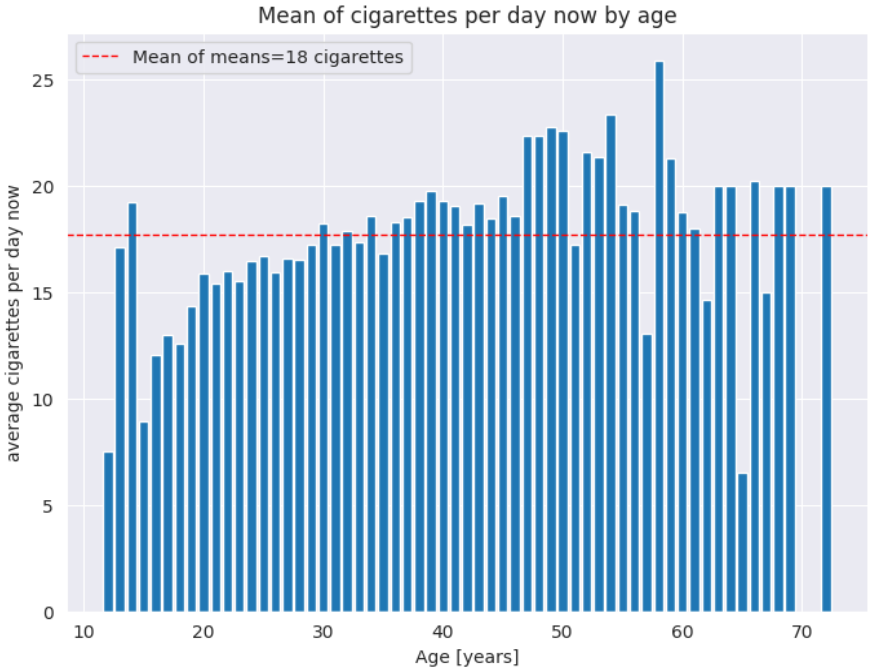

Figure 24: Average number of smoked cigarettes per day given age.

5.2.2 Binary variables

For each pair of the binary variables *pairwise chi-square test of independence* was performed on the *0.05 level of significance*. *Critical value* of the chi-square statistics is in that case 3.841 (see *Table 2*). First the contingency tables are obtained using *pandas.crosstab* function and chi-square statistic and corresponding p-value is computed using scikit-learn implementation of chi square test *scipy.stats.chi2_contingency* function for each pair. The degrees of freedom are calculated by the function automatically, but it would be equal to 1 in this simplest case. Statistically significant correlation was found between *sex* and *using medication regularly.*

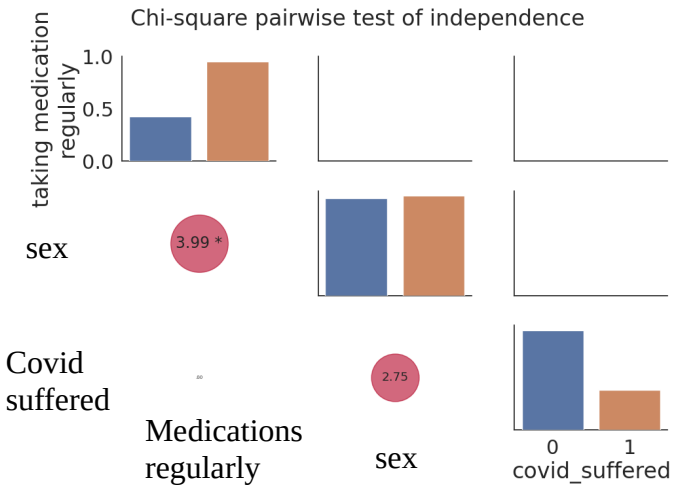

Figure 25: Chi square pairwise test of independence for binary variables. For the dof=1 and significance level equal to 0.05 chi-square critical value for performed chi-square test is 3.841

5.2.3 Ordinal variables

For the ordinal variables were computed Spearman's rank-order correlation coefficient (*Spearman's ρ*) and Kendall's rank correlation coefficient (*Kendall's τ*) (see *Table 2*). Both test found mild negative correlation between the variables *town size* and *educational background* with the coefficient values being equal to *-0.14* for *Spearman's ρ* and *-0.12* for *Kendall's τ* respectively. The town size is however encoded as *1* being the largest and *5* representing the smallest town. This means that there is a positive correlation between how big the town is and highest achieved education level. The bigger the town, the more educated patients, which is somewhat expected result. No correlation between these two variables and *suffered covid* variable was observed.

5.2.4 Testing differences in means for full-dataset split by finished therapy

This section describes experiments, where the full-version dataset was split into two parts by the *finished therapy* variable. Than methods for analyzing whether the means the two populations statistically significantly differ or not. For this purpose could be user 1-way ANOVA, if it's requirements on the data distributions are met. If the data are not normally distributed or if they are not homoscedastic, non-parametric tests like Kruskal-Wallis can be used. First the populations are visualized, later the normality tests is performed and will be find out, that the data are not normally distributed. Than the Kruskal-Wallis test is used, but finds no statistically significant difference. Distributions for the two populations are depicted on *Figure 26:*

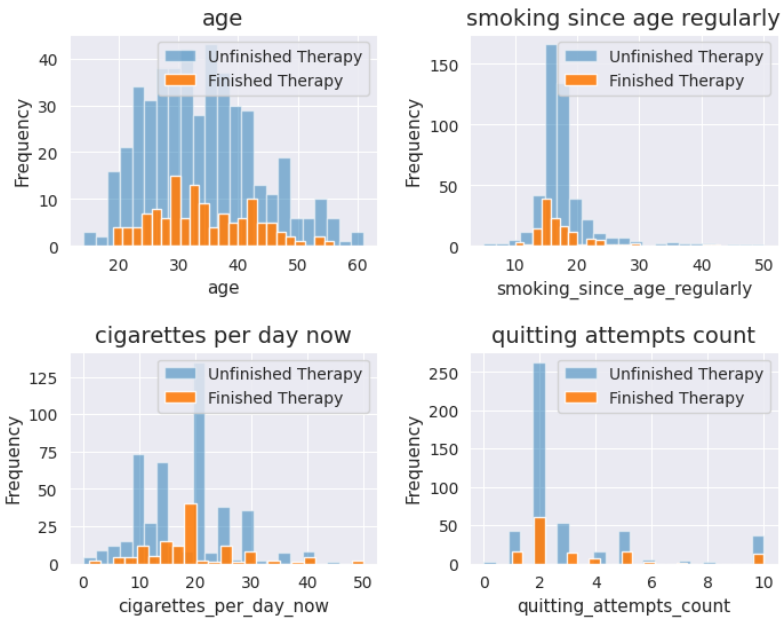

Figure 26: Numeric variables distributions for the full-version dataset split into two parts by 'finished therapy' target variable.

For Shapiro-Wilk normality test we strongly reject the null hypothesis, that the numerical values for all variables, for both population are normally distributed. *Table 12* reports the results:

Table 12: Shapiro-Wilk normality test for numerical variables. Observing the resulting p-values we strongly reject, for the 0.05 level of significance, the null hypothesis, that the data was drawn from a normal distribution, for all variables.

Feature name	p-value finished therapy	p-value unfinished therapy
age	0.31	0.31
smoking_since_age_regularly	0	0
cigarettes_per_day_now	0	0
quitting_attempts_count	0	0

After rejecting that the data come from a normal distribution, Kruskal-Wallis test is performed, on the significance level equal to 0.05, testing the null hypothesis that population medians of the two groups are equal. Observing the results, null hypothesis can be accepted for all the variables. Meaning there is no difference in the mean values for any inspected numerical feature between the two populations. For variables *cigarettes per day now* and *educational background* was almost observed a significant results. *Table 13* reports the results:

Table 13: Kruskal-Wallis *H-test for independent samples, for both ordinal and numerical data. The values reported are the p-values obtained from the test. It can be concluded from the resulting values, that on the 0.05 level of significance, we can accept the null hypothesis. Means do not differ for the two populations of patients who finished therapy and those who did not.*

Feature name	**Resulting p-value**
age	0.54
smoking_since_age_regularly	0.32
cigarettes_per_day_now	**0.07**
quitting_attempts_count	0.39
educational_background	**0.08**
town size	0.96
last_non_smoking_period_duration	0.18

By performing these test we conclude that no significant relationships can be observed just by inspecting some two variables separately. To find more complex relationships and patterns, we proceed using more complex methods.

5.3 Scaling variables

By inspecting the continuous variables distributions and by performing the normality tests, it can be concluded, that the continuous variables are not normally distributed. The proper way to scale those features is the *min-max feature scaling* or what is usually referred to in literature just as a *normalization.*

5.4 Multivariate analysis

This section inspects the relationships between more than two variables, namely between *age*, *educational background* and *sex* – Figure 27 and between s*moked cigarettes per day*, *town size* and *age – Figure 28.* The number of possible combinations of 3 grows fast, so only two significant results are presented and it is rather continued to dimensionality reduction and clustering techniques.

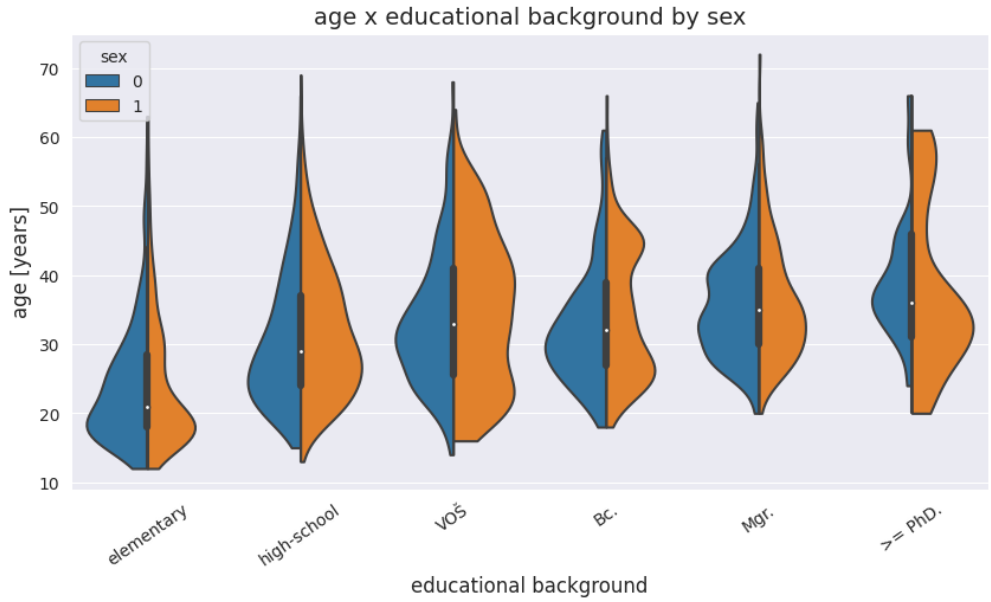

Figure 27: Age vs educational background by sex.

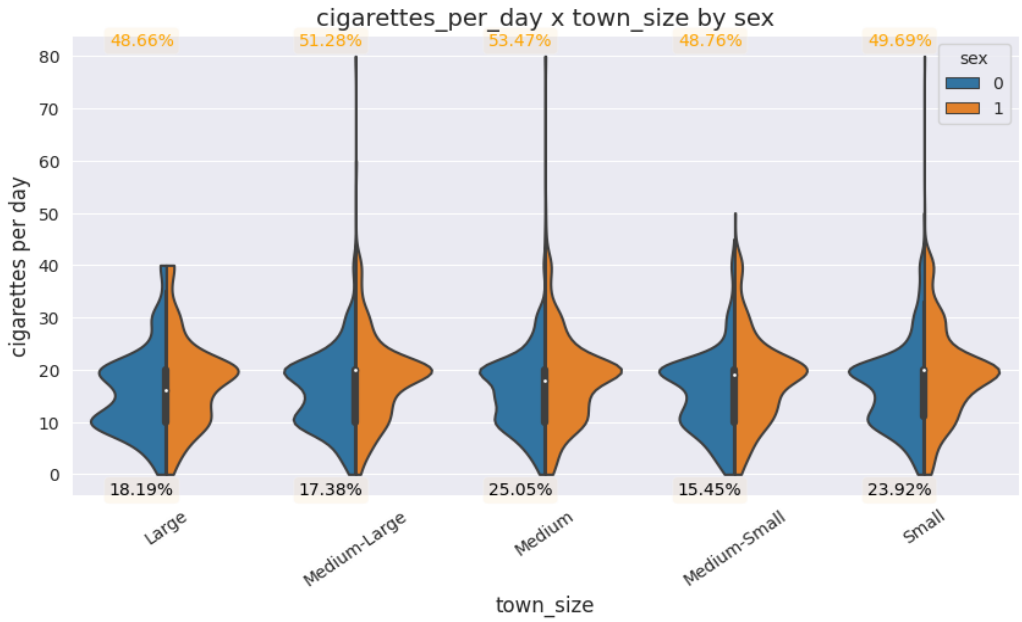

Figure 28: Smoked cigarettes per day vs town size by age.

5.5 Dimensionality reduction

Data are projected into 2 dimensions, using 6 different dimensionality reduction methods, namely: *PCA, Isomap, t-SNE, UMAP, encoders* from *AE* and *VAE*. No elaborate tuning of the hyper-parameters was performed, reasonable defaults, as discussed in Chapter 3 were used. Nonetheless, behavior of the algorithms was thoroughly tested manually on both full-version and combined datasets the results presented in following section shows the best obtained results.

5.6 Clustering

This section summarizes performed clustering experiments on the full-version dataset using the unsupervised clustering methods described in Chapter 3, namely: *agglomerative hierarchical clustering, hdbscan, optics* and *k-means.* The dataset is either directly input into the algorithms, or it's dimensionality is reduced by using *autoencoder,* with latent dimension smaller than the dataset dimensionality, before passing to the algorithm. Arguably the most intriguing step is choosing the number of the clusters. This can be done by optimizing some metric designed for cluster evaluation (*Table 3*). However, computing some score or metric could be fruitless, without thoroughly understanding and inspecting the data. For that reason both approaches are combined. At first, as much as possible knowledge about the dataset is obtained during the exploratory analysis. Then, the data are visualized in two dimensions, using algorithms from distinct dimensionality reduction technique families, to maximize the probability of capturing the real structure of high-dimensional dataset. The data reduced to the two dimensions are then scatter-plotted and colored by the clustering result. This is a great clue not just for setting the hyper-parameters of the algorithms, but also for checking visually, that the data really do cluster.

5.6.1 Setting the hyperparameters

For the descriptive part of evaluation following procedure is used. It produces, from the cluster evaluation metrics point of view, optimal results.

1. For each clustering method: *{hdbscan, optics, hierarchical clustering, k-means}*
2. if reasonable, perform hyper-parameter grid search, aggregating clustering evaluation metrics: S*ilhouette score*, *Calinski-Harabasz index*, *Davies-Bouldin score* each algorithm run. For the *k-means* algorithm also compute the *gap statistics,* choosing the optimal number of clusters.
3. Retrieve some *k* number of optimal hyper-parameters for each used metric.

The results from this step are then used as a clue and a sanity check for manually setting the hyperparameters for each algorithm, by visually inspecting how well the algorithm manages to

find the clusters visible in the 2 dimensional projections, by tweaking the hyperparameters. This has to be done, because lot of optimal hyper-parameters values produces sub-optimal clustering results. Also the 3 different metrics (*Table 3*), tend to have optimums for different hyperparameters. This approach has two premises. Of course, it can be done only if some clusters are present in the projection. Secondly, the manually set hyperparameters give statistics that differ only by some small margin from the ‘optimal’ ones. If the statistics shows that the clustering was successful, inspect the characteristics of the population in the found cluster(s), creating the patients phenotypes.

5.6.2 Clustering algorithms input

As final input to the clustering algorithms, from many tried feature subsets,following featurea set was chosen, with features listed in *Table 14.*

Table 14: Used feature subsets

Features Set A	Features Set B
sex	sex
covid_suffered	covid_suffered
taking_medication_regularly	taking_medication_regularly
age	age
cigarettes_per_day_now	cigarettes_per_day_now
smoking_since_regularly	smoking_since_regularly
quitting_attempts_count	quitting_attempts_count
town_size	town_size
educational_background	educational_background
last_non_smoking_period_duration	last_non_smoking_period_duration
product_using_at_least_once_a_week	
state_of_health	
employment_type	
last_withdrawal_method	

The dimensionality of resulting dataset has size of 594 patients with feature space with 26 dimensions for set A (Entity Embeddings inflate the space a bit) and 10 features for set B.

All numeric features were scaled using the *min-max scaling.* Ordinal variables were scaled to range 0 to 1 with equal step. Binary variables were not scaled. There is a possibility implemented to reduce the weights of the binary variables, but for the following experiments weight was set to 1. The embedding dimensions for the nominal data were set from 3 to 5 according the size of the vocabulary (number of present classes). The deep embedding models were trained repeatedly to convergence.

5.6.3 hdbscan results

Firstly, hdbscan hyperparameters grid search was performed for the minimal size of a cluster (*min_cluster_size)* and for the minimum number of samples in a neighborhood of a point, to be considered a '*core*' point (*min_samples).* Size of the grid was 5 to 100 with step of 5 for both parameters. The differences between the top-3 optimal values are minimal-to-none, for any used metric. *Table 15* presents the results:

Table 15: Optimal values found by the grid search for the set A, with hyperparameter values and metric value in the optimum. Presented results are in form (min_cluster_size, min_samples): metric_value

Rank	S*ilhouette score*	*Davies-Bouldin score*	*Calinski-Harabasz index*
1.	(35, 5): 0.030	(45, 5): 42.680	(45, 5): 2.677
2.	(25, 5): 0.027	(55, 5): 42.680	(55, 5): 2.677
3.	(45, 5): 0.019	(65, 5): 42.680	(65, 5): 2.677

Table 16: Optimal values found by the grid search for the set B, with hyperparameter values and metric value in the optimum. Presented results are in form (min_cluster_size, min_samples): metric_value

Rank	S*ilhouette score*	Davies-Bouldin score	Calinski-Harabasz index
1.	(5, 5): 0.358	(35, 5): 153.064	(5, 95): 1.955
2.	(15, 5): 0.358	(45, 5): 153.064	(15, 95): 1.955
3.	(35, 5): 0.347	(55, 5): 153.064	(25, 95): 1.955

Next, the best hyperparameter values were tried and used as a guide for manually setting the results presented in the plots that follow. Manually set hyperparameters are:

- *set A*: *min_cluster_size=10, min_samples = None*
- *set B: min_cluster_size=15, min_samples = 5*

Each subplot shows the data projected into 2 dimensions colored by labels found the *hdbscan* algorithm. Blue color represent patients who are not assigned to any cluster. **Seven** well-defined clusters were found for set A, *Figure 29 and* **eight** well defined clusters were found for *set B, Figure 30.*

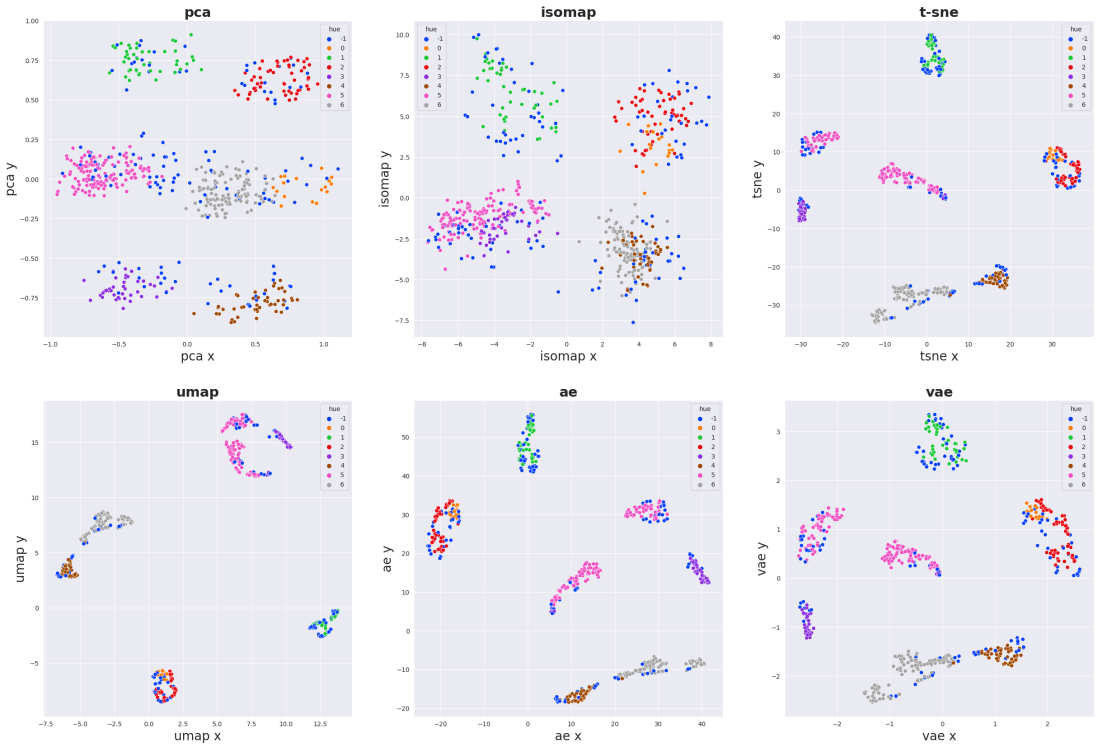

Figure 29: Hdbscan results coloring projections of dataset with feature set A. Seven clusters emerged. Results are visualized using 6 different dimensionality reduction techniques.

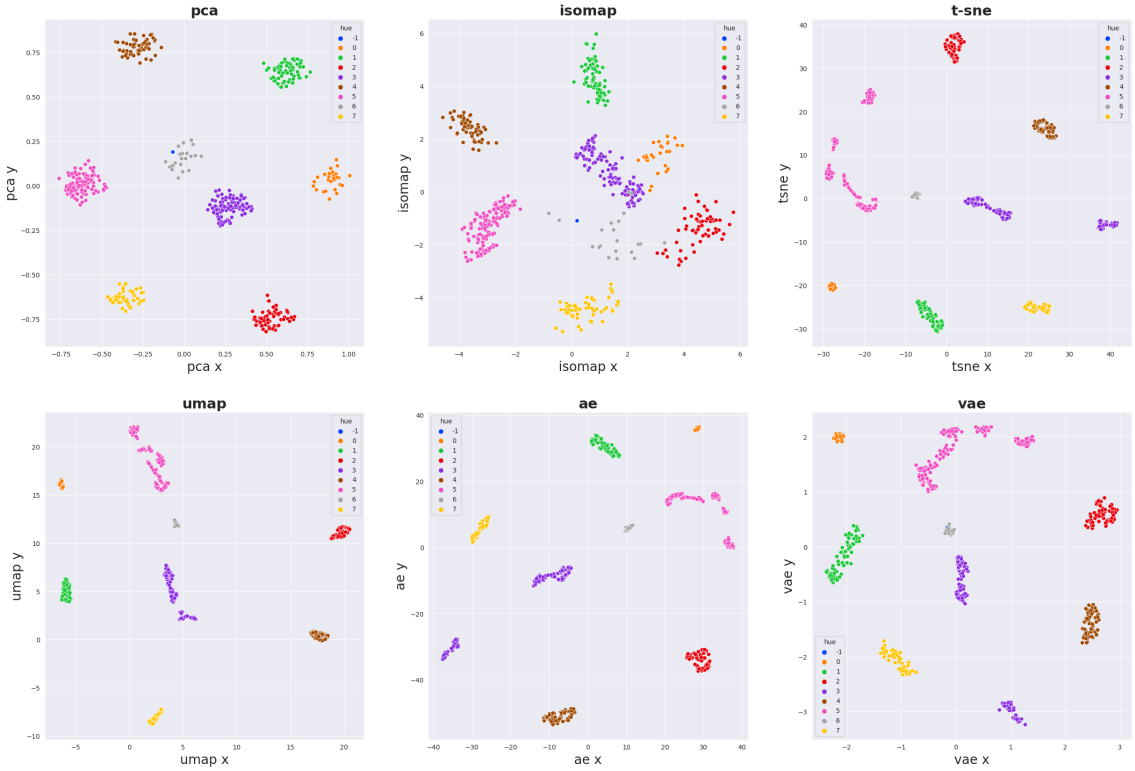

Figure 30: Hdbscan results coloring projections of dataset with feature set B. Eight visible clusters.

5.6.4 Agglomerative hierarchical clustering results

Also the hierarchical clustering was able to find the 7 clusters for set B, *Figures 36, 37*

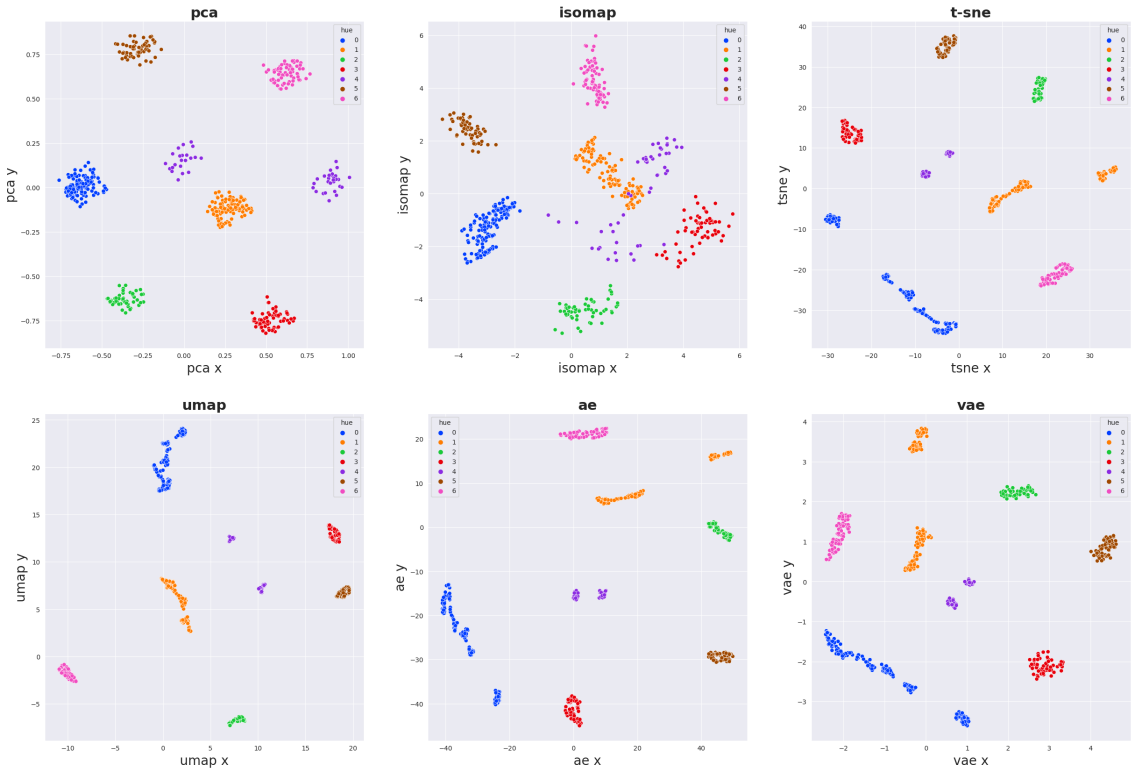

Figure 31: Hierarchical clustering results on set B.

Dendrogram is just another way, how to visualize the clustering results.

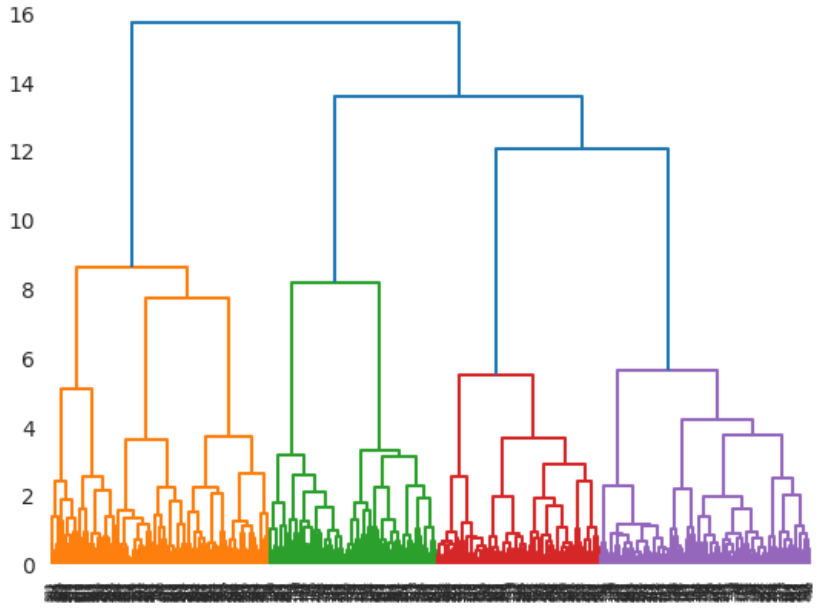

Figure 32: Hierarchical clustering dendrogram for set B

Hierarchical clustering was rather unsuccessful for set A, *Figure 33* and *Figure 34:*

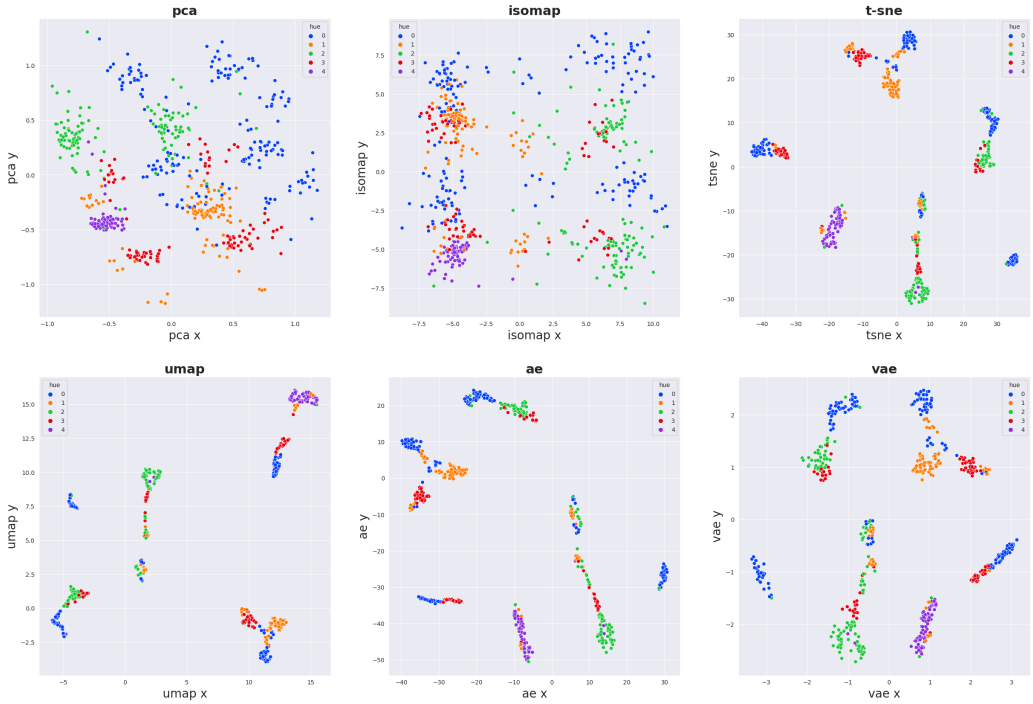

Figure 33: Hierarchical clustering results on set A.

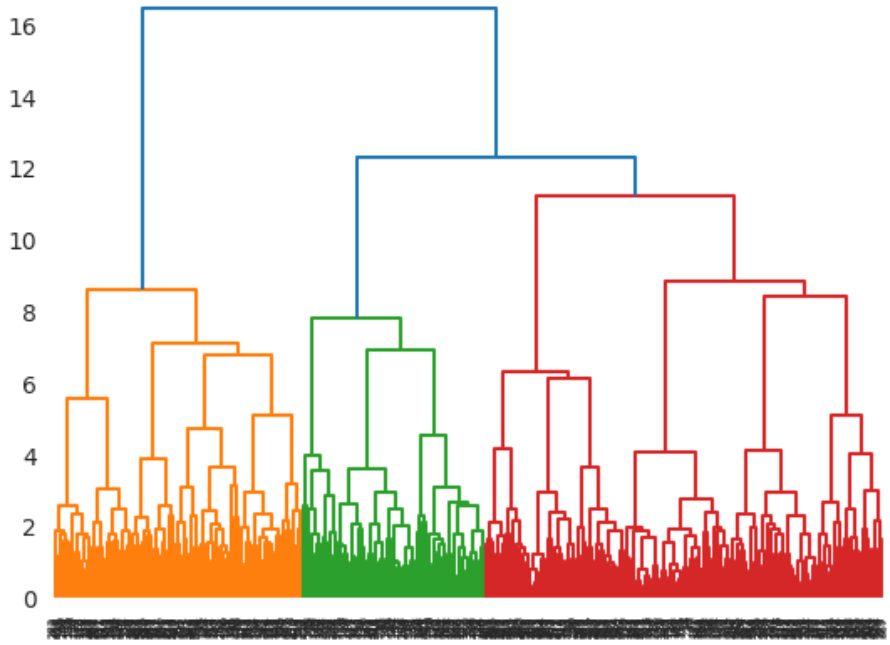

Figure 34: Hierarchical clustering dendrogram for set A.

5.6.5 OPTICS clustering results

For set B optics performs reasonable – *Figure 35*, but for set A it fails to present any results, *Figure 36.*

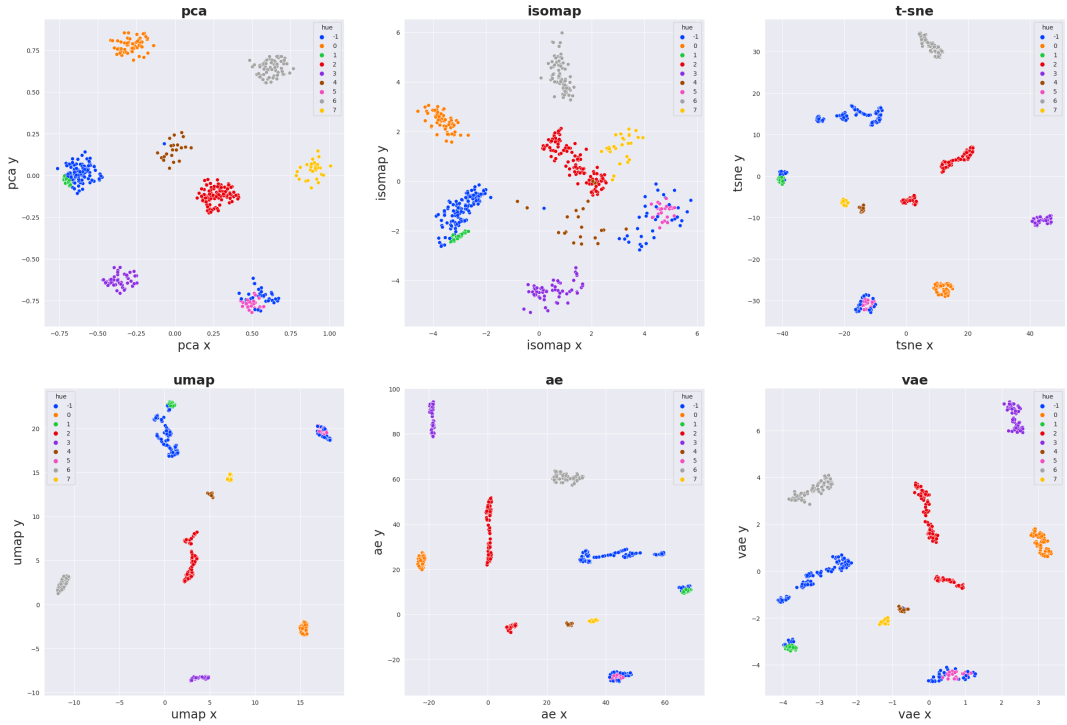

Figure 35: Optics clustering results for the set B.

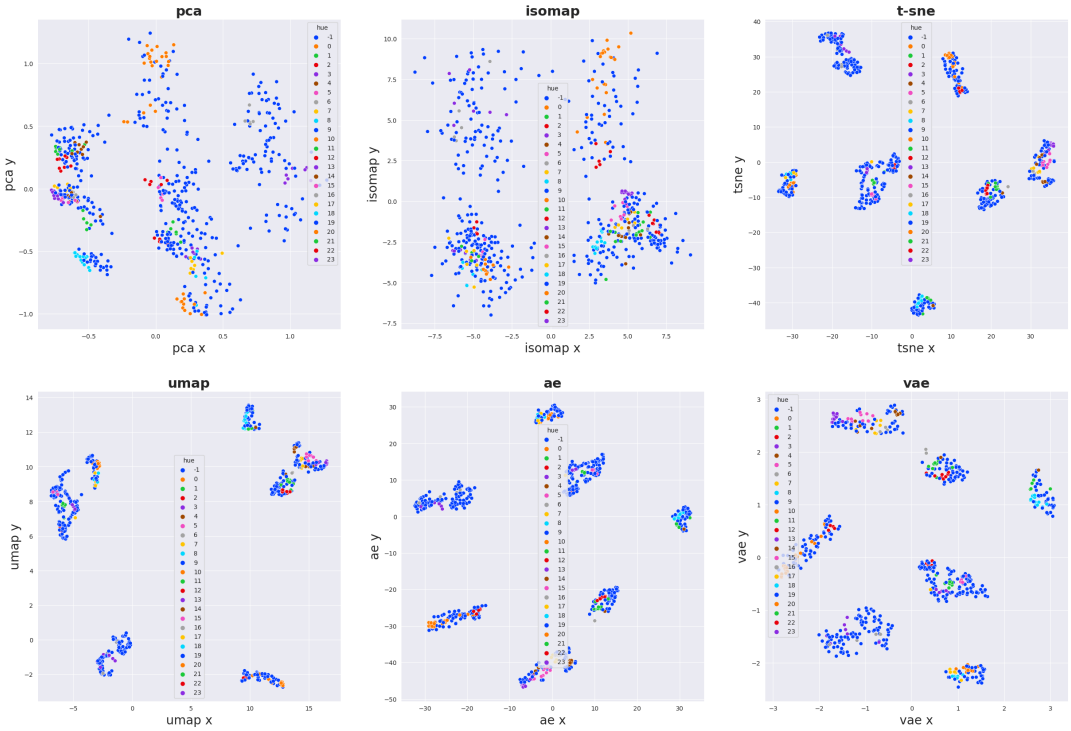

Figure 36: Optics clustering results for set A.

5.6.6 K-means clustering results

Prior to running k-means algorithm, the encoder part of the Autoencoder network trained on the input dataset was used to reduce the dataset dimensionality. It was trained for 1 thousand epochs to convergence, with latent dimension being equal to 4. Once the dimensionality is reduced, k-means algorithm is used to identify clusters in the data. Optimal number of clusters is determined by using the *gap statistics* algorithm and it was equal to 10. However, it was manually set from 10 to 8 by observing the projections. Results for the feature sets A and B are depicted on *Figures 37, 38, 39, 40, and 41*.

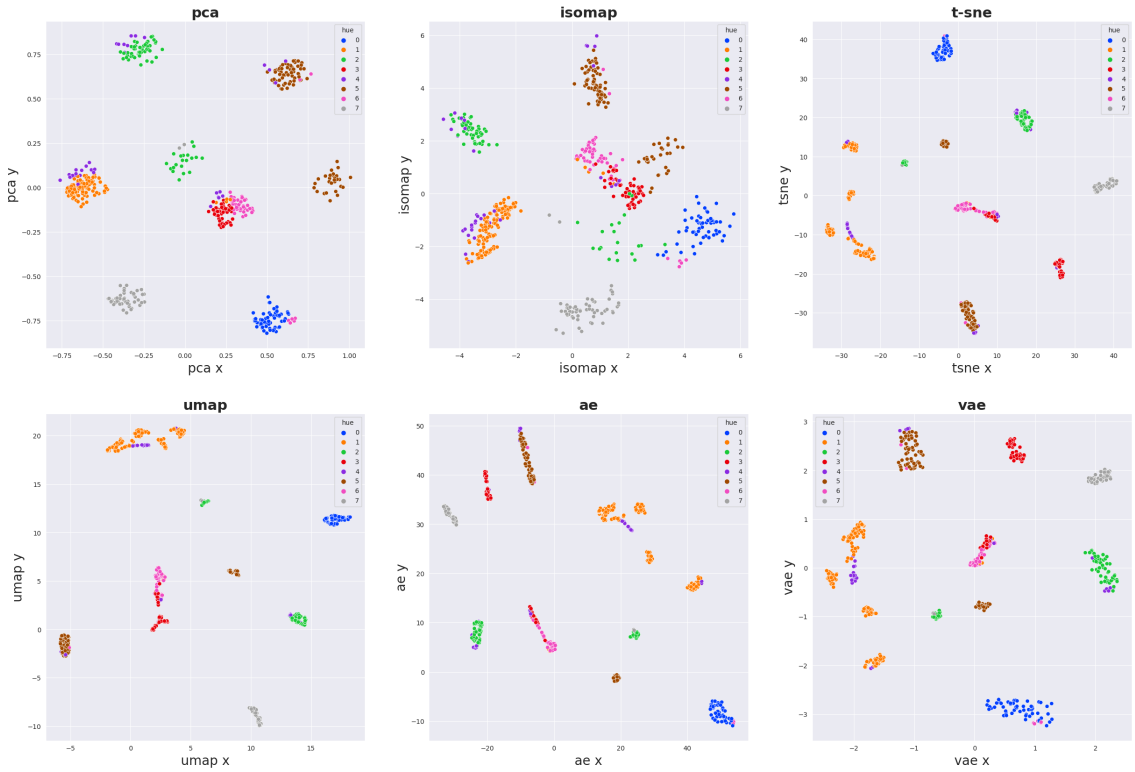

Figure 37: Projected dataset using feature set B and colorized clusters as found by k-means algorithm.

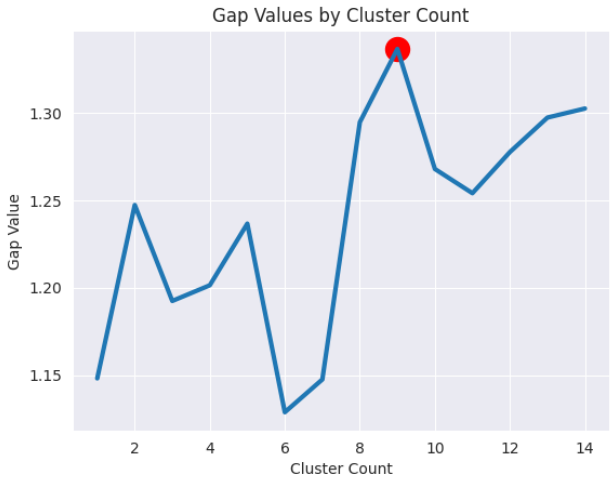

Figure 38: Gap statistic values with optimum at 10 and sub optimum at 8 which was in the end chosen.

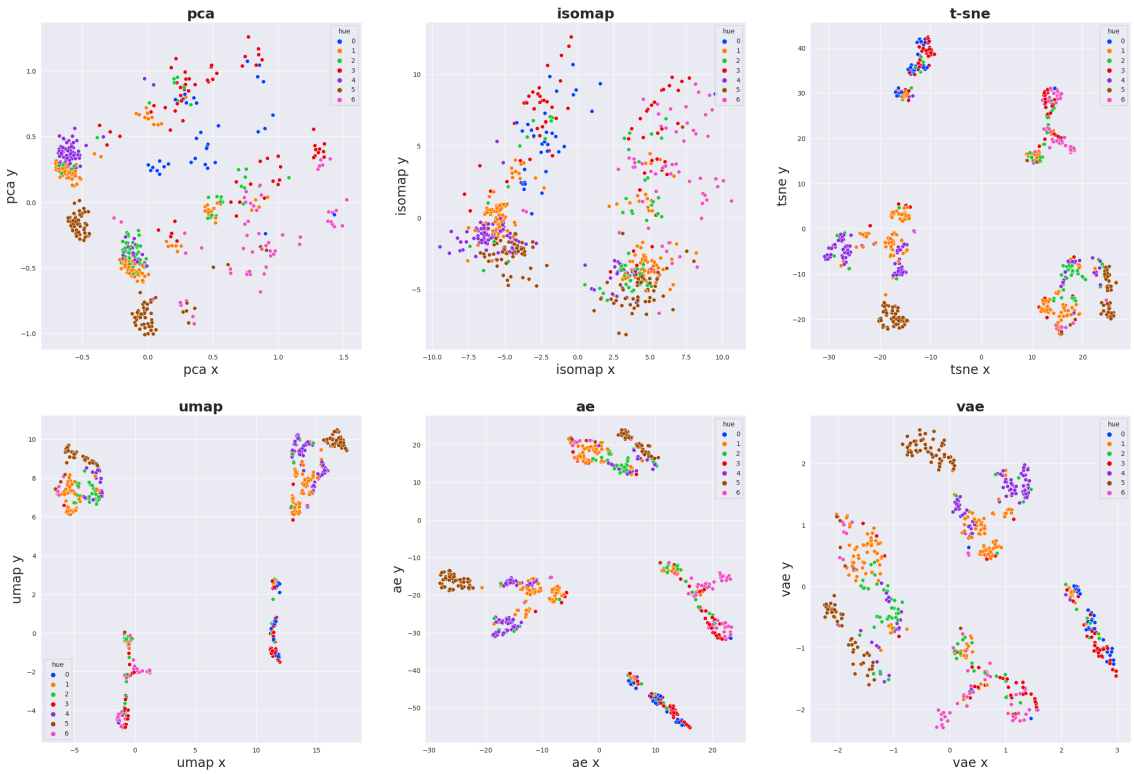

Figure 39: Clusters found by k-means algorithm with using autoencoder first for dimensionality reduction for set A. Seven emerging clusters observed

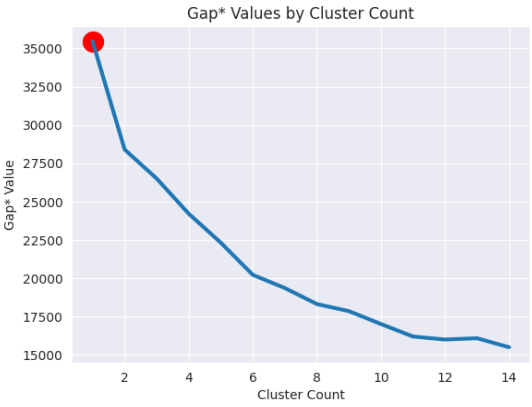

Figure 41: Gap for set A*

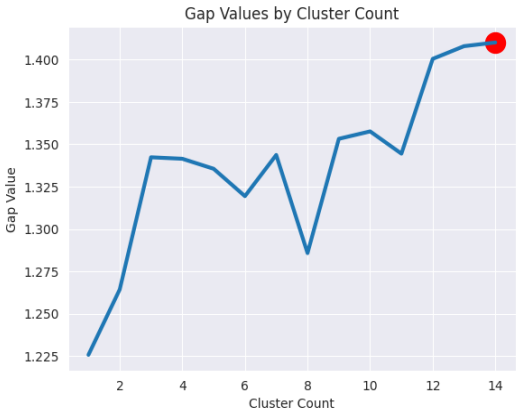

Figure 40: Gap statistic for set A

Gap statistics return 3 optimal value suggestion, using 3 different internal algorithms, presented are just two of them. It can be seen the results are contradictory, the *gap** suggesting only 1 clusters *Figure 34* while the original *gap statistic* method suggests 14, *Figure 35*. However, number of clusters in in *Figure 33* was manually set to 7.

5.6.7 Clustering performance evaluation

All the resulting scores for all algorithms on set A and set B, with according values of clustering performance metrics are presented in the *Table 17* for set *A* and in *Table 18* for *set B.* As noted before, score values for set afterwards by hand do not differ dramatically from the 'optimal' results found by the grid search.

Table 17: Clustering performance evaluation metrics for feature set A.

algorithm	*Silhouette*	*Calinski-Harabasz*	*Davies-Bouldin*	# of found clusters
hdbscan	-0.05	24.46	2.24	7
k-means	**0.19**	**107.85**	**1.64**	7
optics	-0.16	6.54	1.71	x
Agglomerative clustering	0.07	57.09	2.12	7

Table 18: Clustering performance evaluation metrics for feature set B.

algorithm	*Silhouette*	*Calinski-Harabasz*	*Davies-Bouldin*	# of found clusters
hdbscan	**0.36**	128.89	1.13	8
k-means	0.29	**188.71**	1.13	8
optics	0.2	94.71	**1.37**	8
Agglomerative clustering	**0.35**	156.74	1.24	5

For the following presentation of digital phenotypes, the best performing algorithm was chosen.

5.7 Digital phenotypes

The last section of the Results, presents the *digital patients phenotypes* as identified by the used clustering algorithms. Digital phenotypes are computed from the clustered patients features. For the numerical values the computed mean and standard deviation is computed. For ordinal variables is used the median and for binary variables is computed a ratio of the two classes, but in all cases clustering algorithms do not mix the distinct binary classes into one cluster – for example male and female patents were not observed to be assigned to same cluster. For the nominal features can be used three approaches. The first one is to choose the value that is found the most in the given cluster. The second one is to make an intersection of the sets that represent the categories for the given cluster. The third solution would be to report all the observed categories for given nominal variable. Last mentioned approach is used.

5.7.1 Digital phenotypes found by k-means for feature set A.

The phenotypes are listed according to the cluster label ordering on *Figure 33*, it performed the best.

Phenotype 0: Female, 34.4 y.o., smoking since 17.2 y.o. and smoking 17.6 cigarettes per day, with 3.0 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **17.6%**

Phenotype 1: Female, 32.8 y.o., smoking since 17.9 y.o. and smoking 16.9 cigarettes per day, with 2.0 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **20.1%**

Phenotype 2: Female, 35.7 y.o., smoking since 17.5 y.o. and smoking 18.2 cigarettes per day, with 3.7 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **20.3%**

Phenotype 3: Female, 37.3 y.o., smoking since 17.9 y.o. and smoking 19.7 cigarettes per day, with 4.7 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **23.9%**

Phenotype 4: Female, 33.1 y.o., smoking since 17.3 y.o. and smoking 19.7 cigarettes per day, with 4.4 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **24.5%**

Phenotype 5: Female, 34.2 y.o., smoking since 16.9 y.o. and smoking 18.8 cigarettes per day, with 2.8 previous quitting attempts, using medications:No, covid suffered:Yes, resides in town of size=3, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **22.2%**

Phenotype 6: Female, 32.5 y.o., smoking since 18.6 y.o. and smoking 17.2 cigarettes per day, with 2.6 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=2, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **18.6%**

5.7.2 Digital phenotypes for feature set B

Digital phenotypes found by k-means with prior Autoencoder features dimensionality reduction on feature set B (Table 17) . Using smaller feature set, seems to provide finer-grained phenotypes then using the larger feature set A. The results are in the same order and colorized as used in *Figure 31.*

Phenotype 0: Female, 34.8 y.o., smoking since 17.3 y.o. and smoking 17.5 cigarettes per day, with 3.1 previous quitting attempts, using medications:Yes, covid suffered:Yes, resides in town of size=3, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **18.6%**

Phenotype 1: Female, 32.8 y.o., smoking since 17.1 y.o. and smoking 19.0 cigarettes per day, with 2.6 previous quitting attempts, using medications:Yes, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **23.7%**

Phenotype 2: Male, 34.9 y.o., smoking since 18.4 y.o. and smoking 18.6 cigarettes per day, with 2.8 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **14.1%**

Phenotype 3: Female, 31.0 y.o., smoking since 17.0 y.o. and smoking 17.1 cigarettes per day, with 2.7 previous quitting attempts, using medications:Yes, covid suffered:No, resides in town of size=4, with educational background=2, length of previous non-smoking duration:3 || Patients who finished therapy in this cluster : **20.6%**

Phenotype 4: Female, 38.1 y.o., smoking since 17.9 y.o. and smoking 20.2 cigarettes per day, with 9.8 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:2 || Patients who finished therapy in this cluster : **28.9%**

Phenotype 5: Female, 35.0 y.o., smoking since 17.9 y.o. and smoking 17.9 cigarettes per day, with 2.6 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=3, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **18.9%**

Phenotype 6: Female, 34.7 y.o., smoking since 18.9 y.o. and smoking 15.6 cigarettes per day, with 2.3 previous quitting attempts, using medications:No, covid suffered:No, resides in town of size=1, with educational background=4, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **22.0%**

Phenotype 7: Male, 33.6 y.o., smoking since 16.8 y.o. and smoking 19.9 cigarettes per day, with 2.8 previous quitting attempts, using medications:No, covid suffered:Yes, resides in town of size=3, with educational background=3, length of previous non-smoking duration:4 || Patients who finished therapy in this cluster : **26.3%**

For the ordinal value ranges see the univariate analysis results *Figures 15, 16 and 17.*

Chapter 5

Conclusion

Literature and latest methods concerning the digital phenotypes were thoroughly inspected and approach utilizing deep latent models for dimensionality reduction followed by various clustering algorithms was chosen for the solution. Proposed experimental pipeline successfully managed to produce phenotypes of various granularity for the chronic smokers population, depending on the hyperparameters and features used. The quality of resulting clusters was inspected using various metrics for clustering performance evaluation and in most cases two methods prevailed. First one is to use Autoencoder to reduce features dimensionality prior to running k-means. Second approach is using hdbscan. For both methods the data need to be cleaned, scaled and the nominal features encoded using Entity Embeddings. Dataset creation and preprocessing pipeline was constructed along the way, allowing further utilizing the dataset.

Regarding the found digital phenotypes, seemingly the least successful group of patients could be characterized as: a man, 35 years old, who does not use medication on daily bases and did not suffer covid19, resides in middle-sized town and achieved only high-school education. Patients characterized by this values tend to finish therapy the least with only 14% of them finishing the therapy. On the other hand, patients who could be characterized as: a woman, 38 years old, with 10 or more quitting attempts, not using any medication nor suffered covid19, from middle-sized town with high-school education, finished therapy the most, with 29% of patients identified by this phenotype finished the therapy.

In the Introduction was stated, that in order to cluster chronic patients in general, some kind of apparatus or algorithmic pipeline must be created. Both the theoretical and experimental part of this work shows, that there are many possible ways to construct such a pipeline. The one, presented in this work, seems to be a valid. The number of hyper-parameters is still very high, but it was shown that reasonable defaults can be provided. However solving the problem still needs a thorough inspection of the dataset manually and it is preferable to have some prior knowledge about the topic and not just use a plain data-driven approach. Also the found results could be further compared to the known statistics about chronic nicotine patients.

Future improvements follows. The sub-task of classifying the population provided some target variable, in our case it was the therapy success (referred to as '*finished therapy*'), could be viewed also as a task for supervised learning. Also, there could be improvements made for the data gathering process on smartphone application side of the pipeline. The results from this thesis could be used as a feedback for either more effective data gathering or improving the therapy, by focusing more on patients subgroups, utilizing the proposed phenotypes. Such procedure would ideally create a loop, iterating back and forth between the gathering the data from patients and subsequent analysis improving the therapy.

Bibliography

[1] J. Wang et al., “Smartphone Interventions for Long-Term Health Management of Chronic Diseases: An Integrative Review,” Telemedicine and e-Health, vol. 20, no. 6. Mary Ann Liebert Inc, pp. 570–583, Jun. 2014. doi: 10.1089/tmj.2013.0243.

[2] E. Wiecek, A. Torres-Robles, R. L. Cutler, S. I. Benrimoj, and V. Garcia-Cardenas, “Impact of a Multicomponent Digital Therapeutic Mobile App on Medication Adherence in Patients with Chronic Conditions: Retrospective Analysis,” Journal of Medical Internet Research, vol. 22, no. 8. JMIR Publications Inc., p. e17834, Aug. 12, 2020. doi: 10.2196/17834.

[3] S. D. Birkhoff and S. C. Smeltzer, “Perceptions of Smartphone User-Centered Mobile Health Tracking Apps Across Various Chronic Illness Populations: An Integrative Review,” Journal of Nursing Scholarship, vol. 49, no. 4. Wiley, pp. 371–378, Jun. 12, 2017. doi: 10.1111/jnu.12298.

[4] H. K. Ubhi, D. Kotz, S. Michie, O. C. P. van Schayck, and R. West, “A comparison of the characteristics of iOS and Android users of a smoking cessation app,” Translational Behavioral Medicine, vol. 7, no. 2. Oxford University Press (OUP), pp. 166–171, Feb. 06, 2017. doi: 10.1007/s13142-016-0455-z.

[5] N. F. BinDhim, K. McGeechan, and L. Trevena, “Smartphone Smoking Cessation Application (SSC App) trial: a multicountry double-blind automated randomised controlled trial of a smoking cessation decision-aid ‘app,’” BMJ Open, vol. 8, no. 1. BMJ, p. e017105, Jan. 2018. doi: 10.1136/bmjopen-2017-017105.

[6] P. Portelli and C. Eldred, “A quality review of smartphone applications for the management of pain,” British Journal of Pain, vol. 10, no. 3. SAGE Publications, pp. 135–140, Jul. 08, 2016. doi: 10.1177/2049463716638700.

[7] J.-P. Onnela, “Opportunities and challenges in the collection and analysis of digital phenotyping data,” Neuropsychopharmacology, vol. 46, no. 1. Springer Science and Business Media LLC, pp. 45–54, Jul. 17, 2020. doi: 10.1038/s41386-020-0771-3.

[8] B. L. Haskins, D. Lesperance, P. Gibbons, and E. D. Boudreaux, “A systematic review of smartphone applications for smoking cessation,” Translational Behavioral Medicine, vol. 7, no. 2. Oxford University Press (OUP), pp. 292–299, May 19, 2017. doi: 10.1007/s13142-017-0492-2.

[9] K. Regmi, N. Kassim, N. Ahmad, and N. Tuah, “Effectiveness of Mobile Apps for Smoking Cessation: A Review,” Tobacco Prevention & Cessation, vol. 3, no. April. E.U. European Publishing, Apr. 12, 2017. doi: 10.18332/tpc/70088.

[10] J. Webb et al., “Long-Term Effectiveness of a Clinician-Assisted Digital Cognitive Behavioral Therapy Intervention for Smoking Cessation: Secondary Outcomes From a Randomized Controlled Trial,” Nicotine & Tobacco Research, vol. 24, no. 11. Oxford University Press (OUP), pp. 1763–1772, Apr. 26, 2022. doi: 10.1093/ntr/ntac113.

[11] J. Torous, M. V. Kiang, J. Lorme, and J.-P. Onnela, “New Tools for New Research in Psychiatry: A Scalable and Customizable Platform to Empower Data Driven Smartphone Research,” JMIR Mental Health, vol. 3, no. 2. JMIR Publications Inc., p. e16, May 05, 2016. doi: 10.2196/mental.5165.

[12] J.-P. Onnela, “Opportunities and challenges in the collection and analysis of digital phenotyping data,” Neuropsychopharmacology, vol. 46, no. 1. Springer Science and Business Media LLC, pp. 45–54, Jul. 17, 2020. doi: 10.1038/s41386-020-0771-3.

[13] G. Fagherazzi, “Deep Digital Phenotyping and Digital Twins for Precision Health: Time to Dig Deeper,” Journal of Medical Internet Research, vol. 22, no. 3. JMIR Publications Inc., p. e16770, Mar. 03, 2020. doi: 10.2196/16770.

[14] K. Huckvale, S. Venkatesh, and H. Christensen, “Toward clinical digital phenotyping: a timely opportunity to consider purpose, quality, and safety,” npj Digital Medicine, vol. 2, no. 1. Springer Science and Business Media LLC, Sep. 06, 2019. doi: 10.1038/s41746-019-0166-1.

[15] M. Loi, “The Digital Phenotype: a Philosophical and Ethical Exploration,” Philosophy & Technology, vol. 32, no. 1. Springer Science and Business Media LLC, pp. 155–171, Jun. 11, 2018. doi: 10.1007/s13347-018-0319-1.

[16] S. Shiffman, M. S. Dunbar, S. M. Scholl, and H. A. Tindle, “Smoking motives of daily and non-daily smokers: A profile analysis,” Drug and Alcohol Dependence, vol. 126, no. 3. Elsevier BV, pp. 362–368, Dec. 2012. doi: 10.1016/j.drugalcdep.2012.05.037.

[17] S. K. Kachigan, Statistical analysis. Radius Press, 1986. isbn: 0942154991, 9780942154993

[18] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065. The Royal Society, p. 20150202, Apr. 13, 2016. doi: 10.1098/rsta.2015.0202.

[19] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” Science, vol. 290, no. 5500. American Association for the Advancement of Science (AAAS), pp. 2319–2323, Dec. 22, 2000. doi: 10.1126/science.290.5500.2319.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008.

[21] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.” arXiv, 2018. doi: 10.48550/ARXIV.1802.03426.

[22]J. B. Kruskal, “Nonmetric multidimensional scaling: A numerical method,” Psychometrika, vol. 29, no. 2. Springer Science and Business Media LLC, pp. 115–129, Jun. 1964. doi: 10.1007/bf02289694.

[23] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” Science, vol. 313, no. 5786. American Association for the Advancement of Science (AAAS), pp. 504–507, Jul. 28, 2006. doi: 10.1126/science.1127647.

[24] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes.” arXiv, 2013. doi: 10.48550/ARXIV.1312.6114.

[25] T. J. Loftus et al., “Phenotype clustering in health care: A narrative review for clinicians,” Frontiers in Artificial Intelligence, vol. 5. Frontiers Media SA, Aug. 12, 2022. doi: 10.3389/frai.2022.842306.

[26] R. Gove, L. Cadalzo, N. Leiby, J. M. Singer, and A. Zaitzeff, “New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation,” Visual Informatics, vol. 6, no. 2. Elsevier BV, pp. 87–97, Jun. 2022. doi: 10.1016/j.visinf.2022.04.003.

[27] Ö. Akay and G. Yüksel, “Clustering the mixed panel dataset using Gower’s distance and k-prototypes algorithms,” Communications in Statistics - Simulation and Computation, vol. 47, no. 10. Informa UK Limited, pp. 3031–3041, Dec. 06, 2017. doi: 10.1080/03610918.2017.1367806.

[28] M. Wattenberg, F. Viégas, and I. Johnson, “How to Use t-SNE Effectively,” Distill, vol. 1, no. 10. Distill Working Group, Oct. 13, 2016. doi: 10.23915/distill.00002.

[29] Z. I. Kalantan and N. A. Alqahtani, “A study of principal components analysis for mixed data,” International Journal of ADVANCED AND APPLIED SCIENCES, vol. 6, no. 12. International Journal of Advanced and Applied Sciences, pp. 99–104, Dec. 2019. doi: 10.21833/ijaas.2019.12.012.

[30] J. J. McArdle, “Latent Variable Modeling of Differences and Changes with Longitudinal Data,” Annual Review of Psychology, vol. 60, no. 1. Annual Reviews, pp. 577–605, Jan. 01, 2009. doi: 10.1146/annurev.psych.60.110707.163612.

[31] A. Creswell and A. A. Bharath, “Denoising Adversarial Autoencoders.” arXiv, 2017. doi: 10.48550/ARXIV.1703.01220.

[32] A. Makhzani and B. Frey, “k-Sparse Autoencoders.” arXiv, 2013. doi: 10.48550/ARXIV.1312.5663.

[33] S. Gu and L. Rigazio, “Towards Deep Neural Network Architectures Robust to Adversarial Examples.” arXiv, 2014. doi: 10.48550/ARXIV.1412.5068.

[34] A. Asperti and V. Tonelli, “Comparing the latent space of generative models.” arXiv, 2022. doi: 10.48550/ARXIV.2207.06812.

[35] Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, “On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression,” IEEE Signal Processing Letters, vol. 27. Institute of Electrical and Electronics Engineers (IEEE), pp. 1485–1489, 2020. doi: 10.1109/lsp.2020.3016837.

[36] X. Bao, J. Lucas, S. Sachdeva, and R. Grosse, “Regularized linear autoencoders recover the principal components, eventually,” arXiv, 2020, doi: 10.48550/ARXIV.2007.06731.

[37] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering,” The Journal of Open Source Software, vol. 2, no. 11. The Open Journal, p. 205, Mar. 21, 2017. doi: 10.21105/joss.00205.

[38] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “OPTICS,” ACM SIGMOD Record, vol. 28, no. 2. Association for Computing Machinery (ACM), pp. 49–60, Jun. 1999. doi: 10.1145/304181.304187.

[39] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms.” arXiv, 2011. doi: 10.48550/ARXIV.1109.2378.

[40] M. Capó, A. Pérez, and J. A. Lozano, “An efficient K -means clustering algorithm for massive data.” arXiv, 2018. doi: 10.48550/ARXIV.1801.02949.

[41] Ren et al., “Deep Clustering: A Comprehensive Survey.” arXiv, 2022. doi: 10.48550/ARXIV.2210.04142.

[42] A. Ahmad and S. S. Khan, “Survey of State-of-the-Art Mixed Data Clustering Algorithms,” IEEE Access, vol. 7. Institute of Electrical and Electronics Engineers (IEEE), pp. 31883–31902, 2019. doi: 10.1109/access.2019.2903568.

[43] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms.” arXiv, 2011. doi: 10.48550/ARXIV.1109.2378.

[44] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the Number of Clusters in a Data Set Via the Gap Statistic,” Journal of the Royal Statistical Society Series B: Statistical Methodology, vol. 63, no. 2. Oxford University Press (OUP), pp. 411–423, Jul. 01, 2001. doi: 10.1111/1467-9868.00293.

[45] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2. Institute of Electrical and Electronics Engineers (IEEE), pp. 224–227, Apr. 1979. doi: 10.1109/tpami.1979.4766909.

[46] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” Communications in Statistics - Theory and Methods, vol. 3, no. 1. Informa UK Limited, pp. 1–27, 1974. doi: 10.1080/03610927408827101.

[47] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” Journal of Computational and Applied Mathematics, vol. 20. Elsevier BV, pp. 53–65, Nov. 1987. doi: 10.1016/0377-0427(87)90125-7.

[48] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, “Explaining the Performance of Multi-label Classification Methods with Data Set Properties.” arXiv, 2021. doi: 10.48550/ARXIV.2106.15411.

[49] C. Guo and F. Berkhahn, “Entity Embeddings of Categorical Variables.” arXiv, 2016. doi: 10.48550/ARXIV.1604.06737.

Appendix A

Attachments structure

This appendix section describes the implementation structure of the code used for all the experiments.

koleckar /

+ *jupyter_notebooks /*
  * *clustering /*
    - *hdbscan.ipynb*
    - *hierarchical_clustering.ipynb*
    - *k_means.ipynb*
    - *optics.ipynb*
  * *correlations /*
    - *correlations_binary.ipynb*
    - *correlations_ordinal.ipynb*
    - *correlations_continuous.ipynb*
    - *corrections_mixed.ipynb*
    - *all_vs_finished_therapy.ipynb*
  * *visualizations /*
    - *visualization_categorical_variables.ipynb*
    - *visualization_dimension_reduction.ipynb*
    - *visualization_numeric_variables.ipynb*
    - *visualize_data_cleaner_steps.ipynb*
    - *visualization_multiple_options_variables.ipynb*
    - *statistics_users_under_30.ipynb*
+ *postgresql_queries /*
  * *retrieve_users_data.sql*
  * *get_unique_features.sql*
+ *logs /*
+ *graphs /*
+ *ae.py*
+ *vae.py*
+ *categorical_embeddings.py*
+ *data_cleaner.py*
+ *dimred.py*
+ *plotting_functions.py*

(+ *users_data.pkl*)

+ requirements.txt
+ *README.md*

The code was written with strong emphasis on clean code princips, enabling possibile future extensibility. The API is also fairly well documented for the classes. However, the jupyter (*.ipynb*) notebooks have more of an experimental and visualization character.