TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

Defining Corporate Health Classes

Authors	Anastasia Makarevich, Ion Barbu, Min Wu, Moritz Müller
Mentor(s)	M.Sc. Jan Kukačka (Helmholtz Zentrum)
	Dr. Dominik Jüstel (Helmholtz Zentrum)
	Dr. Sebastian Dünnebeil (wellabe)
Co-Mentor	M.Sc. Konstantin Göbler
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Feb 2021

Abstract

The medical field requires insights based on machine learning models, but in many cases there is not enough labelled data available. One way to address this problem is to use an external dataset to compensate for missing information and train on it, while using a local dataset to test on. The problem of this approach is that, although both datasets come from the general human population, they can be quite different due to a sample selection bias. This leads to a mismatch in the joint distribution of the two datasets, which is often referred to as *dataset shift*. In this project, we investigate the consequences of dataset shift as well as research methods for fixing it. These methods include naive methods like resampling nearest neighbors, and more sophisticated ones like Kullback-Leibler Importance Estimation Procedure and Boosted Decision Tree reweighter. We use the American National Health and Nutrition Examination Survey dataset as our external rich source and a local dataset from the Munich start-up wellabe as our test set.

We evaluate our results on a regression task for age prediction and several classification tasks. We conclude that in most cases reweighting methods provide moderate improvement on the target metric in the test set. We also show that reweighting cannot be considered as a universal tool and its power is limited by the task at hand: rare diseases transfer worse, while easier tasks with strong predictors transfer better.

Contents

1 Introduction

In recent years, companies have become more interested in employees' health due to the rapid changes in working style and working environment. wellabe, a start-up based in Munich, aims to provide its client companies and their employees better understanding of their health status as well as personalized prevention programs.

After a 15-minute on-site check-up taken place directly at the client companies, an employee's health profile containing various biomarkers is built. The check-up includes measuring blood pressure for heart health assessment, taking blood samples to analyse metabolic processes, evaluating breath for lung volume and pulmonary conditions, estimating body composition values via a smart scale, and lastly measuring heart rate variability to analyse stress levels [1]. One's medical history as well as lifestyle-related behaviors are also collected in a questionnaire during this process: it is recorded, for example, if the person has diabetes. After the check-up, the results are interpreted and explained by doctors via video consultations. Health scores are then given to the individuals based on their overall health condition. In the end, personalized prevention programs, which detect early-stage risk factors, are recommended to the employees via the wellabe mobile app, so that better precautionary measures can be taken to build a healthier lifestyle and possibly prevent diseases.

The health data collected during check-ups can be used to build machine learning models and thus improve our understanding of the relationships between diseases and their related factors. However, there are several challenges that make it a hard task. Firstly, due to privacy concerns, one cannot directly access the client data, so in this project we had to use a synthetic dataset, which mimics the underlying distributions of the real wellabe data. The synthetic dataset imposes many limitations and challenges to work with due to its synthetic nature, for example, creating erroneous small clusters of values, or not being able to capture the true correlations between features. Secondly, both real and synthetic data do not contain sufficient amount of labels to build disease prediction models, only diabetes is available, so we need an external source of data. For these reasons, another dataset, namely the National Health and Nutrition Examination Survey dataset (NHANES), sampled based on the United States population over nearly 20 years, is introduced with a rich set of biomarkers and other additional health labels.

The problem with such an approach is that although these two datasets come from the same population (all people on earth), they represent different subsets of this population, which we can refer to as *sample selection bias*. Specifically, NHANES dataset represents the general United States population with large diversity, whereas the wellabe dataset only represents the German corporate class, in which employees' health is cared for by the companies. This selection bias leads to another problem, known as *dataset shift* [2], which generally means that the joint distributions of features x and target y, $p(x, y)$, are different in the two datasets. For us, it means that we cannot directly apply the models trained on NHANES to the wellabe dataset.

Our main challenge in this project was to address the aforementioned dataset shift problem, and explore if we can alleviate it to make NHANES models applicable to wellabe. We first researched the different types of dataset shifts, which include covariate shift, prior probabilities shift, and concept shift. After evaluating the severity of shift between NHANES and wellabe, we explored various resampling strategies to fix this issue. There-

fore, we started with simple methods like resampling the nearest neighbours, and then investigated more complex methods like Kullback-Leibler Importance Estimation Procedure (KLIEP) and Boosted Decision Tree reweighting (BDT).

We evaluated our approach on a simple regression task for age prediction and also on a variety of classification tasks. As we have mentioned, wellabe dataset does not contain enough labels, so we introduced additional artificial features by discretizing some of the available continuous ones. We concluded that our reweighting stategies gave moderate improvement in most cases based on the target evaluation metric. We also discovered several limitations for model transfer for the cases when the classification task is difficult and cannot be easily explained by the data at hand. Namely, the task of predicting diabetes became a challenging one when there were many missing values for glucose level measurements. However, even for such tasks, there was a significant improvement after applying our reweighting strategies. Thus, we concluded that model transfer is possible but the outcomes highly depend on the chosen task.

Lastly, we also performed several experiments with NHANES data to see which particular labels and features could be of potential use for wellabe and can be easily included. The first of such tasks was asthma prediction for which we conclude that wellabe can benefit from including the current medication and bronchitis disease status of a person. Then we also trained another model on NHANES to detect if the person has taken medication in the last 30 days. Knowing this information is very important and can help to get additional insights when interpreting check-up results. In the end, we also suggested a method for wellabe to validate the quality of this model on the wellabe dataset.

The project report is structured as follows. We first introduce the datasets and available features in Chapter 2. Next, in Chapter 3 we show domain analysis and explain our cleaning and preprocessing procedures. Chapter 4 is dedicated to the problem of dataset shift and our approaches to alleviating it. In the next two Chapters 5 and 6, we apply the methods from Chapter 4 for regression and classification modeling tasks and analyse the results. Finally, Chapter 7 shows how we can apply the model transfer results for working with labels that are not available in the real wellabe dataset.

2 Datasets

2.1 wellabe Datasets

The wellabe dataset, measured on German employees as explained in Chapter 1, will be referred to in the succeeding sections as "real wellabe". This particular dataset contains 70 features and 1,500 samples. Those features are separated into five groups. Four of them are the main feature groups containing numerical biomarkers categorized based on their medical meanings: cardiovascular system, metabolism, respiratory system, and body composition. The fifth and the last group includes additional features for personal information (e.g., age and sex), medical conditions (e.g., if one has diabetes), lifestyle habits and behaviors (e.g., if one ate or exercised recently before the check-up), health scores (e.g., scores based on doctor's consultation), recheck status and related features (e.g., if a follow-up check-up is conducted), and bookkeeping labels (e.g., user id,

record of when the appointment is booked, etc.).

Given, that samples come from the corporate environment, real wellabe only contains people of age from 18 until 65, and is not considered to be representative for the entire German population.

To protect patient privacy and follow data protection laws, a synthetic version of the wellabe dataset was used for modeling purposes in this project instead. The synthetic wellabe dataset is an algorithm-generated dataset which mimics real wellabe and maintains the underlying feature distributions. It contains 50,000 samples and 70 features (same features as in real wellabe). There are two main advantages for using such a synthetically-generated dataset. Firstly, the anonymity of the patients is kept. The synthetic data is generated by creating new samples instead of introducing noise to the original data, making it harder to retrieve sensitive personal information for individuals. Secondly, the size of the synthetic dataset is much larger than real wellabe, allowing us to build more robust and generalizable models. However, working with synthetic data introduced a lot of challenges, which will be further explained in Section 2.3.

The primary functionality of the synthetic wellabe dataset in this project was used for modeling and experimenting various methods introduced in Chapter 4. The final model transfer evaluation was based on the performance tested on the real wellabe dataset, which was hidden from the students.

2.2 NHANES Dataset

The motivation of this project is to make use of a richer dataset compared to wellabe, specifically, NHANES. NHANES is a survey and research program built upon multiple studies aiming to assess the health and nutritional status of the American population [3]. The dataset includes a combination of interviews and health examinations of survey participants. The samples are recorded in two-year buckets since the year 1999, making NHANES a rich complementary dataset to the smaller wellabe dataset with only 1,500 samples.

The initial NHANES version provided for this project contained 28 features, which were also included in wellabe, and roughly 72,000 samples for the years 1999-2012. During the course of this project, we decided to enrich the NHANES dataset and include the years 2013-2018 in order to make use of the additional 30,000 samples. To validate the enrichment, we investigated equipment used, lab methods, and lab site used during the medical examinations for each of the year buckets. Except for the feature for glucose level, we could not find any significant changes. Therefore, all of the other features had the same medical meanings and were included for the additional years.

One issue with the NHANES dataset was that not all features were present in all year buckets (see Appendix A), creating many missing values for certain features. The problem will be addressed by our imputation procedure explained in Section 3.2.2.

For this project, the NHANES dataset was used as the training set for our models. Given its richness compared to the wellabe dataset, we aim to not only transfer models trained on NHANES to real wellabe, but also predict target variables which are not measured by wellabe.

2.3 Limitations of Synthetic Data

Synthetic datasets, generated by computer simulation, have many limitations, especially in complex systems as the human body. The challenge is to accurately model not only separate distributions, but also the complex relationships between random variables.

As a result, the wellabe synthetic dataset naturally posed challenges in terms of medical inconsistencies and general data inconsistencies. In-depth research was conducted and initial data analysis was done on all the features in order to guide the cleaning process. This thorough research was crucial to understand which health features needed to be adjusted, or which features could not be used when testing a model trained on a separate dataset. When considering the features that could be recomputed such as the Fatty Liver Index (FLI) and Body Mass Index (BMI) (see Appendix B for detailed feature explanation and calculation), the values in the synthetic dataset would not match our recalculated value. Moreover, certain medical inequalities were violated, for example, many samples had a higher forced expiratory volume in 1 second (FEV1_L) than their total forced expiratory volume (FVC_L).

Initial data analysis of the synthetic dataset showed small clusters at outlier values. One example of this is seen in high-density lipoprotein (HDL), the graph in Figure 1 shows approximately 400 samples with an HDL level of 124 mg/dL, which is a substantial amount for this outlier value. On the lower end of HDL values, many samples have values clustered in the range of 1 to 9 mg/dL. This posed challenges because more in-depth knowledge was required to determine if those values were valid or not, the domain was restricted to exclude the outlier values.

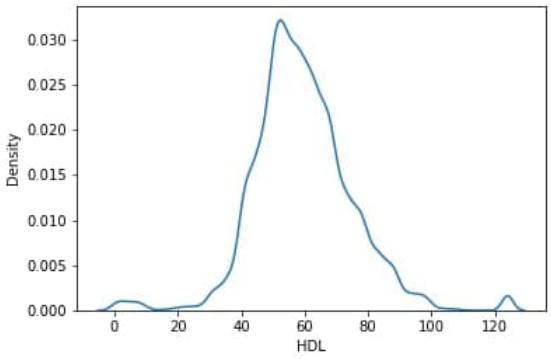

Figure 1: Distribution with small clusters

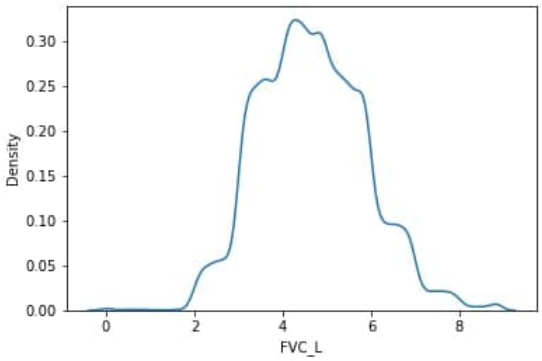

Figure 2: Non-smooth distribution

Moreover, many features follow non-smooth density distributions. Figure 2 shows an example pattern for FVC_L. In this case, the density distribution of this feature has large jumps and drops seen at FVC_L values of 3 liters and 6 liters. These jumps and drops limit the ability to accurately estimate the true distribution, which is more normally distributed.

Lastly, although the synthetic dataset captures individual features' distributions as in real wellabe, correlations between features are not always maintained. Features that should be strongly correlated due to their medical meanings, for example, alanine and aspartate transaminases (ALT, AST), have correlation of 0.77 in NHANES, but only 0.18 in the synthetic dataset.

3 Domain Understanding and Data Handling

3.1 Domain Understanding

Using various online medical research sources based out of universities [4] and clinics [5], as well as Centers for Disease Control and Prevention [6], detailed exploration on all the features of the datasets was performed. Further research was done on each of the features to determine which values were considered healthy. This so-called "normal range" was defined as a range of values that correspond to the ideal values for that label. Comparing the values in the datasets against those normal ranges provided us medical understanding of the features along with insights about the individuals' health conditions. The normal ranges can be found in Table 9 under Appendix B, alongside the corresponding feature descriptions.

All the features were categorized into one of the five groups: metabolism, respiratory system, cardiovascular system, body composition, and other features which did not fit into the previous four groups. Table 10 in Appendix C shows how each feature was categorized and its availability in the wellabe and NHANES datasets.

3.2 Data Cleaning and Handling

3.2.1 General Cleaning Procedure

We examined each feature in both datasets in detail and implemented a thorough cleaning procedure. The data cleaning rules were divided into 3 subgroups: rules that only apply to the NHANES or the wellabe dataset, and rules that are applicable for both. A comprehensive list of the most relevant used cleaning procedures can be found in Table 11 in Appendix D.

For both datasets, several general rules were applied. Zero values were firstly converted to NaNs before the imputation steps. Features that existed in both datasets were compared and cross-checked to ensure they had the same units and thus on the same ground for comparison. Furthermore, we unified encodings (e.g., gender and pregnancy) and data types for features available in both datasets. Lastly, a few additional features were created which could be beneficial for the further modeling tasks. For example, BMI was created based on weight and height for both datasets; Mean Arterial Pressure (MAP) as well as Pulse Pressure (PP) were calculated for NHANES since they were not recorded previously (the exact formulas can be found in Appendix B).

Provided that the samples were collected from different populations, the age groups and their distribution for the two datasets also vary, as shown in Figure 3. NHANES covers a wide age group from infants until age 85, whereas in the corporate wellabe dataset, samples' ages range strictly from 18 to 65. In order to model the two populations as close as possible, while not losing too many rows, we included samples from the NHANES dataset with age from 18 until 75, with the assumption that samples with age 66 - 75 have similar health data as people who are, for example, above 60. This resulted in having around 53% of the original data for NHANES. This age distribution comparison after cleaning can be found in Figure 4.

Cleaning medical data can be a challenging task for various reasons. Questionable (i.e., too high or too low) values in the datasets could be caused by measurement errors

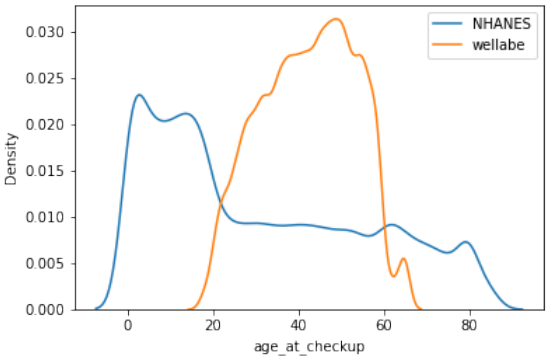

Figure 3: Before cleaning

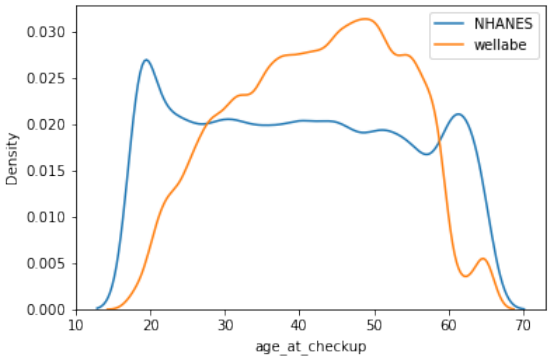

Figure 4: After cleaning

during examinations, algorithmic errors produced by the synthetic data, or they could indicate a certain illness in samples, which require adequate medical understanding to recognize and differentiate.

As already discussed in Section 2.3, after taking an in-depth look at all the features, rules were set in place to handle the issues synthetic dataset generates, such as the small clusters. Moreover, some outlier samples, which were considered as physically impossible, were removed. For example, the normal range for BMI is from 18.5 to 24.9 for both males and females. Samples with extreme BMI values, greater than 62.87 (more than 5 standard deviations away from the mean), were considered as severely obese and thus taken out from NHANES.

3.2.2 Missing Value Imputation

Missing data can be classified into one the the three categories introduced by Rubin [7], describing the different underlying mechanisms that generate the missing data: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). MCAR implies that the missingness of a variable is purely random, and is not related to any other observed or unobserved covariates. MAR suggests the missingness is still random, but is related to some other observed variables. MNAR, on the other hand, refers to the case when the missingness relates only to the unobserved variables, or data that is missing. For example, people with depression might more likely refuse to complete survey questions about depression, creating missingness that is MNAR.

Overall, one should consider missing types when doing imputations and avoid imputing variables that are MNAR, since more biases could be introduced to the estimations. Unfortunately, there exists no well-defined statistical method to determine exactly which category a variable belongs to. Therefore, to reduce potential induced bias, we only imputed variables that had less than 5% missing values.

Since NHANES' features is a subset of wellabe's features and we would only model on NHANES, imputing missing values for features which only existed in wellabe was considered unnecessary. The set of features that appeared in both datasets with less than 5% missing values made up the so-called "usable features" group, which contained the following features: ALT, AST, BMI, CHOLESTEROL, CREA, DIA_BP, GGT, HDL, HEART_RATE, MAP, PP, SYS_BP, WAIST_SIZE, WEIGHT, age_at_checkup, has_diabetes,

height, is_pregnant, and sex (see Appendix B for further feature descriptions). All features were imputed directly, besides has_diabetes, where the missing values were filled as 0 (non-diabetic).

We experimented imputing missing values using k-nearest neighbors (kNN) with various numbers of neighbors. In the end, we chose k = 4 as it gave better performance when modeling for the age prediction task mentioned in Chapter 5, while all other model parameters were kept consistent.

4 Model Transfer

One of the challenges of this project was to research if we can potentially transfer a model trained on NHANES data to wellabe dataset without direct re-training. This kind of setting would be of particular interest for predicting the probability for diseases that have labels in NHANES but not in wellabe. Naturally, one should not expect that the predictions for NHANES will automatically work in wellabe. The problem we have been challenged with is to see if there exists a way to somehow make the knowledge learned from NHANES more applicable to wellabe.

Broadly speaking, the term *dataset shift* describes the setting, when the joint distributions $p(x, y)$ of features x and targets y at the time of training and testing are not the same. This kind of problem happens quite often in real world, when the system developed in lab conditions has to be applied in real world. The problem is that machine learning algorithms often come with the assumption that train and test data come from the same distribution and this is what we cannot guarantee in our scenario with the NHANES and wellabe datasets. The consequence of that is that good performance of the model in the stationary environment (when train and test come from the same distribution) does not guarantee the model will still be valid when dataset shift happens.

One of the major reasons for that is sample selection bias [2]. For example, the wellabe dataset is not representative of the whole German population. However, it is a representative subset of the German corporate class. NHANES at the same time attempts to be representative of the United States population and includes much richer variety of observations, although this richness sometimes can also be a source of problem. For example, the research shows 54% of African American adults have high blood pressure while for Caucasian adults it is just 46% [8].

We can potentially de-bias the train dataset by excluding certain ethnicities, however, the ethnicity label is not available in wellabe and unfortunately it does not solve the problem: even though de-biasing slightly improve the situation with the heavy tails for NHANES, the densities are still different as seen in Figure 5. That is why we had to come up with more sophisticate methods.

With this in mind it is clear that it is very important to address this situation in this project. In the next section we will briefly look at different types of dataset shifts which might occur when switching from train to test setting, namely: covariate shift, prior probability shift and concept shift.

In this and the following section we define x as as set of features or covariate, y as a target variable and $p(x, y)$ as a joint distribution over features and targets.

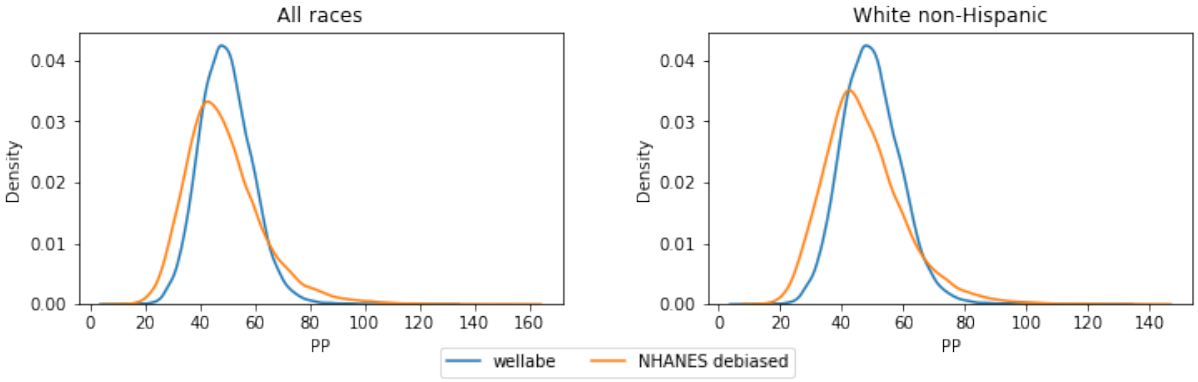

Figure 5: Pulse pressure before and after de-biasing

4.1 Covariate Shift

One of the most well-known types of dataset shifts is the covariate shift. As one can guess from the name, this is the situation when the distribution of covariates is different for train and test datasets, namely [2]:

$$p_{NH}(x) \neq p_{wel}(x)$$

Here, $p_{NH}(x)$ and $p_{wel}(x)$ are the joint density distributions of the features for NHANES and wellabe respectively. The important assumption we are making in this case is that the conditional distributions, $p_{NH}(y|x)$ and $p_{wel}(y|x)$ are equal, which means that. This is important because Amos Storkey [9] raises the idea that if we are able to learn the true conditional model, that could explain the data, then there is no need to compensate for the covariate shift and the accuracy of the predictions should not change much. However, if the test data is simpler than train, then, by compensating for the covariate shift and, for example, resampling, we are allowing for a simpler model, which is not necessarily a true model, but it provides a better performance due to locality.

We should note that we can also end up in the situation when test data is more complex than train and in this case the performance of the model with compensated covariate shift might be even worse.

Another case when matching the distributions might lead to worse results is when the test data we are matching to does not capture the actual relationships in the data. By resampling or reweighting we might end up in the situation where we simply give too much weight to points that are likely to come from test data but are not informative enough for building a good model. As a result, by matching the joint distribution, we might actually break important correlations that are present in real world and captured by train data but that are not present in test due to different reasons as shown in Section 2.3. One notable example are the features of ALT and AST. The correlations between ALT and AST in NHANES is 0.76, while in the synthetic dataset it is just 0.18. After resampling the correlation between these features is 0.6 in resampled version and 0.59 in the case of reweighting. We can observe a similar behaviour with other features. The results are similar both for resampled and reweighted.

We can conclude that under the situation of the covariate shift we can either try to learn the true model or match the distributions such that we can use a simpler model.

However, in both cases we have to keep in mind the assumption that $p_{NH}(y|x) = p_{wel}(y|x)$.

4.1.1 Measuring Covariate Shift

In case of wellabe and NHANES we can see (again, under assumption that $p_{NH}(y|x) = p_{wel}(y|x)$) that the covariate shift is indeed present), which can be seen in Figure 6, where the blue and orange density distributions are from the NHANES and wellabe datasets, respectively.

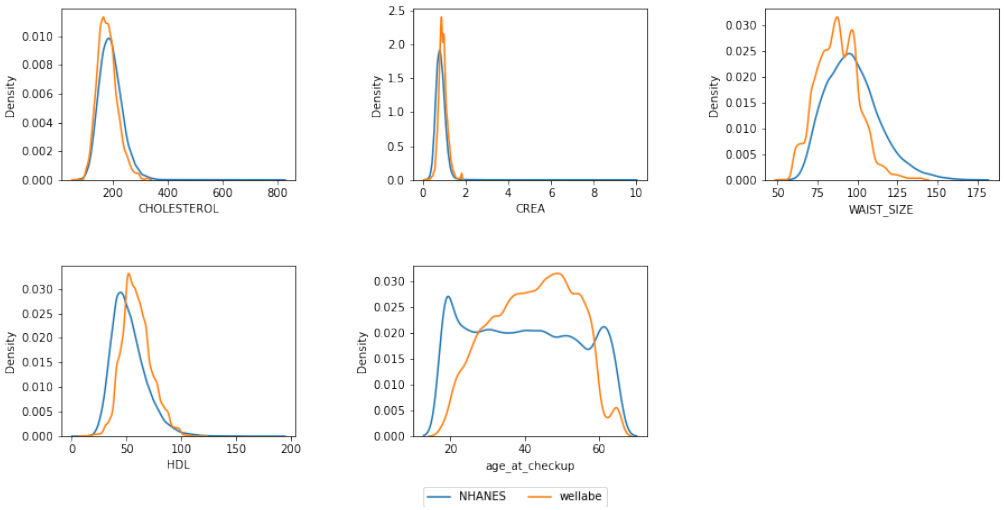

Figure 6: Original distributions of selected features

Before we dive into methods for fixing the covariate shift, it is reasonable to develop a metric, with which we can measure if the suggested method indeed provides an improvement. One way to do so is visual examination, as seen in Figure 6. Another approach would be to train a classifier to distinguish between elements in the two datasets and check its prediction performance. If no covariate shift is present, we should expect the classifier to have a low performance. The algorithm for that can be summarized as follows:

1. Assign 0 to all samples in NHANES and 1 to all samples in wellabe
2. Train a classifier to predict if the sample comes from wellabe dataset, using selected features (the features choice depends on the task for which we are fixing the covariate shift)
3. Evaluate the performance

For performance evaluation, we will the concordance statistic (c-statistic), representing the area under the Receiver Operating Characteristic curve, which is often used to evaluate classification methods. If it is close to 1, it means that the datasets are far apart and we can clearly predict if the sample comes from NHANES or wellabe. As a classificator we used boosted decision tree in all cases. Another metric that we have used to evaluate the covariate shift is Kolmogorov-Smirnov Statistic for each feature [10]:

$$D_n = \sup_{x \in \mathbb{R}} |F_{1,n}(x) - F_{2,m}(x)|,$$

where $F_{1,n}(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x}$ and $I_{X_i \leq x}$ indicates whether observation X_i is in the region $(-\infty, x]$. $F_{2,m}$ is defined in the same way. The KS statistic checks what the largest gap between the two distributions is. The functions $F_{1,n}(x)$ and $F_{2,n}(x)$ check quantile by quantile and find the difference of their cummulative distributions. It can be seen from the definition that the smaller the value of the KS statistic indicate bigger similarity between the distributions.

4.1.2 Methods for Covariate Shift

When the covariate shift is present, ideally one should retrain the model on the new data. However, it is not always possible. In our case specifically, the available train data is richer than test. In the next sections we show different methods for alleviating the covariate shift and evaluate them using the metrics, described above. The summary of these results can be found in Section 4.1.3

4.1.2.1 Naive Methods

One intuitive thing to do in a situation of a covariate shift is to drop the features that are the most problematic and demonstrate the biggest mismatch - for example, age at check-up. However, if we keep removing features, we will end up with a very limited set of covariates that provide less insight to us.

Another idea that comes to mind is to resample the dataset by choosing only similar observations: for each observation x in wellabe, we find observation y in NHANES that is closest in terms of Euclidean distance. In such, we are left with the following optimization problem:

$$y = \arg \min_{y \in NH} \sum_{i}^{d} (y_i - x_i)^2,$$

where given a fixed observation x, we find the closest observation y across d features. This method is very heavy computationally due to necessity to compute all pairwise distances, which results in runtime complexity of $\mathcal{O}(nm \times d)$ because we compute $n = |wellabe|$ distances to $m = |NHANES|$ observations across $d = dim(wellabe)$ features. We will refer to this approach as nearest neighbor resampling (NN).

4.1.2.2 Dataset Prediction Classifier

A lof of methods for the covariate shift rely on the idea that it can be fixed by reweighting the training data according to the ratio of the train ($p_{NH}(x)$) and test ($p_{wel}(x)$) probability distributions. Formally known as the importance estimator [11], these weights are calculated using:

$$w(x) = \frac{p_{wel}(x)}{p_{NH}(x)}$$

Similar to the algorithm described in 4.1.2, we used a dataset prediction classifier (LogReg) to estimate the importance. We start again by assigning 0 to samples in NHANES

and 1 to samples wellabe. Using the probability that $p_{wel}(\sigma = 0)$ and $p_{NH}(\sigma = 1)$, we take this ratio as a an estimate for the importance, where σ denotes the dataset label for NHANES or wellabe. Then the logistic regression model is fitted to predict $p(\sigma|x)$. Then the estimate of the densities ratio can we rewritten as [12]:

$$\frac{p_{wel}(x)}{p_{NH}(x)} = \frac{p(\sigma = 0)}{p(\sigma = 1)} \left(\frac{1}{p(\sigma = 0|x)} - 1 \right)$$

The original methods suggests to incorporate the weighting procedure directly into the learning algorithm. However, we are only using the weights.

4.1.2.3 Kullback Leibler Importance Estimation Procedure

Another method for estimating the density ratio and finding the importance was suggested by Masashi Sugiyama [13] and is called Kullback-Leibler Importance Estimation procedure (KLIEP). The key idea of this approach is to model the importance as a linear combination of basis functions $\xi_i(x)$ scaled by parameters $\{\alpha_i\}_{i=1}^k$, which are learnt from data [13]:

$$\widehat{w}(x) = \sum_{i=1}^{k} \alpha_i \xi_i(x),$$

and then use it to estimate p_{wel}:

$$\widehat{p}_{wel}(x) = \widehat{w}(x) p_{NH}(x)$$

KLIEP uses the Kullback-Leibler divergence (KL divergence), which measures how different two probability distributions are. By taking the formal definition of the KL divergence, we can minimize the distance from p_{wel} to $\widehat{p}_{wel}$, and in the process, obtain an estimate for $\widehat{w}(x)$.

After solving directly from the definition of KL-divergence and empirically approximating this integral, we are left with a concave optimization problem with the objective function to be maximized defined as [13]:

$$\sum_{j=1}^{n_{wel}} \log \left(\sum_{i=1}^{k} \alpha_i \xi_i(x_j^{wel}) \right),$$

subject to $\sum_{l=1}^{n_{NH}} \sum_{i=1}^{k} \alpha_i \xi_i(x_l^{wel}) = n_{NH}$ and $a_i \geq 0 \forall i$. The variables to be optimized are $\{\alpha\}_i$ with a set of initialized basis functions $\xi_i(x)$

This objective function has constraints that can be derived from the fact that $\widehat{p}(x)$ has a density equal to 1. In deriving these constraints, we approximate that integral empirically based on of the testing input (wellabe). Finally, by solving this concave problem with respect to $\{\alpha_i\}_{i=1}^k$, KLIEP allows to estimate the importance ratio $w(x)$ without directly computing the densities.

4.1.2.4 Boosted Decision Tree Reweighter

Another interesting approach that we have found during our research was proposed by Alex Rogozhnikov [14] and is based on boosted decision trees (BDT). The idea is somewhat similar to that of dataset prediction explained in Section 4.1.2.2 . We are again working with the merged dataset, consisting of NHANES and wellabe data with additional label indicating which dataset the observation comes from. In this case, however, we encode wellabe as 0 and NHANES as 1 instead.

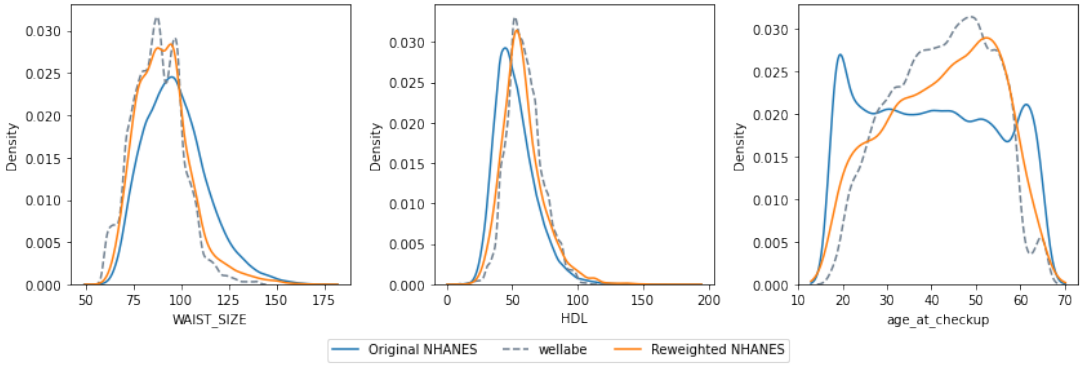

Figure 7: Reweigthing results with BDT

The key idea is to use the decision trees to find the regions (aka leaves) of the feature space of this full dataset that are important for the reweighting. In this tree, the importance is measured with respect to the number of observations from train and test data: if the leaf contains much more data points from train than from test, then the importance of this region should be smaller. In order to achieve this we build many shallow decision trees, where the following metric is maximized (referred to as symmetrizied χ^2 on page 3 in the original paper [14]):

$$\chi^2 = \sum_{leaf} \frac{(|NH_{leaf}| - |wel_{leaf}|)^2}{|NH_{leaf}| + |wel_{leaf}|},$$

where $|NH_{leaf}|$ and $|wel_{leaf}|$ is the number of train (NHANES) and test (wellabe) observations in the leaf respectively. When there is equal number of observations in the leaf, the χ^2 will obviously be zero - we don't have to do anything there. Intuitively it makes sense to ideally have a balance of both datasets in the leaves. After the tree is found the prediction, $\widehat{y}$, is computed using:

$$\widehat{y} = \log(|te_{leaf}|) - \log(|tr_{leaf}|)$$

At each iteration, the weight vector ω is updated using the following formula:

$$\omega_{new} = \omega_{old} \exp(y \cdot \widehat{y}),$$

where y is the true value (1 for NHANES and 0 for wellabe), and the weight vector ω is initialized as a vector of ones. It should be intuitively clear that if the observation y comes from the test dataset (0 for wellabe), then the weight does not change. The runtime for

this method is $\mathcal{O}(m \times p \times n_{trees})$, where m is the size of the combined dataset, p is the number of features and n_{trees} is the number of trees. This runtime can be adjusted when choosing the depth of the decsion tree.

Figure 7 shows the results of the BDT reweighting method, where we can see the original covariate shift for the features: WAIST_SIZE, HDL and age_at_checkup. After reweighting, it is visually clear that the reweighted NHANES follows a similar distribution to wellabe.

4.1.3 Covariate Shift Reweighting Results

In order to compare the covariate shift reweighting methods, we used the average KS statistic for the features ALT, CHOLESTEROL, CREA, GGT, HDL, WAIST_SIZE, height and age_at_checkup. These features were the most complete overlapping between the NHANES and wellabe datasets, and the effects of the covariate shift was clearly seen when no reweighting metod was applied due to a high C-statistic (0.9627), which is close to 1 and high KS statistic (0.27426). The table below shows all of the results for the covariate shift methods mentioned in this section, and it is clear that each method lowered the C-statistic and significantly lowered the KS statistic. The best performing methods were KLIEP and BDT.

Covariate Shift Reweighting method	C-statistic	KS Statistic
No reweighting	0.9627	0.27426
Nearest Neighbour reweighting method	0.8927	0.1217
Dataset prediction method (LogReg)	0.9217	0.2083
KLIEP reweighting method	0.8590	0.128
BDT reweighting method	0.8563	0.0787

Table 1: C-statistic and KS Statistic for different reweighting methods to address covariate shift

4.2 Prior Probability Shift

In contrast to the covariate shift, this type of shift is detected in the target variable y, when its distribution in train and test datasets are different. Using the notation consistent in this section, we have [15]:

$$p_{NH}(y) \neq p_{wel}(y)$$

under the same assumption that $P_{wel}(x|y) = P_{NH}(x|y)$, where $p_{wel}(x|y)$ and $p_{NH}(x|y)$ are the conditional distributions of the features x given label y. This can be the case with data such as diseases, where this target variable is not distributed similarly. For example in NHANES, the percentage of people having diabetes is 8%, while in the synthetic wellabe dataset it is 0.8%, which is around ten times smaller. Comparing these percentages to official census data showed that in United States, 13.3% [16] of age 18-65 have diabetes, while in Germany this percentage is 15.3% [16]. This however, includes both diagnosed

and undiagnosed cases, while we only have data for diagnosed cases (when the patient is aware of their disease and are able to communicate it). So in this case, 8-10% for NHANES seems reasonable. The difference we observe between NHANES and synthetic wellabe might be due to the selection bias: i.e. we know that wellabe users belong to the corporate class and expectantly have a better health. This kind of situation might become problematic for generative models like Naive Bayes, where we rely on priors distribution $p(y)$. For this kind of models there exist methods for adjusting the priors with respect to the new (test) data.

There are two possible settings in this situation: when $p_{wel}(y)$ is known and when it is unknown. In case when it is known, we can adjust the output probabilities of the classifier by using [15]:

$$P_{wel}(y_i|x) = \frac{\frac{p_{wel}(y_i)}{p_{NH}(y_i)} p_{NH}(y_i|x)}{\sum_{j=1}^{n} \frac{p_{wel}(y_j)}{p_{NH}(y_j)} p_{NH}(y_j|x)}$$

When new priors are unknown it is still possible to estimate them for which there are several approaches to do so. One simple approach is the confusion matrix approach which is composed of true and false positives as well as true and false negatives [15]. Applying this kind of adjustments for probabilities of the Naive Bayes improves recall, but severely distorts accuracy, so eventually we made a decision not to adjust the priors. Applying this kind of adjustment for conditional model (in contrast to generative) is questionable and we have not found sufficient support for such methods in literature.

4.3 Concept Shift

Lastly, we would like to address the problem of concept shift. This happens when the density distributions for the features are equal, $p_{NH}(x)$=$p_{wel}(x)$, but [2]:

$$p_{NH}(y|x) \neq p_{wel}(y|x)$$

Simply speaking, the relationships between covariates and targets are different in train and test datasets. This often happens when there is a latent variable we are not aware of. A typical example is when some quantity depends on time of the year, but this variable is not present in dataset and we do not account for it. In our case we can suspect that there are certain features that might have different meaning in NHANES and wellabe. One such example is blood pressure. It was detected that, for example, systolic blood pressure has different relationship to age in the United States compared to Germany, which one can observe in Figure 8.

There are several methods for fixing the concept shift which are mostly targeted at time series data, where this problem naturally occurs. Like with all other shifts, it would be ideal to retrain the model, but we don't have such option. When it comes to non-series data our options are rather limited and mostly reduce again to reweighting or resampling methods [17] that help to filter the train datast and select only relevant samples (which we can do using the methods from Chapter 4).

In Figure 8, we illustrate the effects of relieving the concept shift with the example of systolic blood pressure and age at check-up. Each graph shows the relationship of the

two features in wellabe and different subsamples of NHANES. These following NHANES versions are shown:

1. The raw NHANES dataset.
2. Debiased data: we removed ethnicities that are not representative of the German population and removed people taking medicine as it could be used to regulate blood pressure.
3. Reweighted data: We used the strategies shown in Section 4.1.
4. Reweighted and de-biased: combined approaches of (2) and (3) by first de-biasing and then reweighting the data.

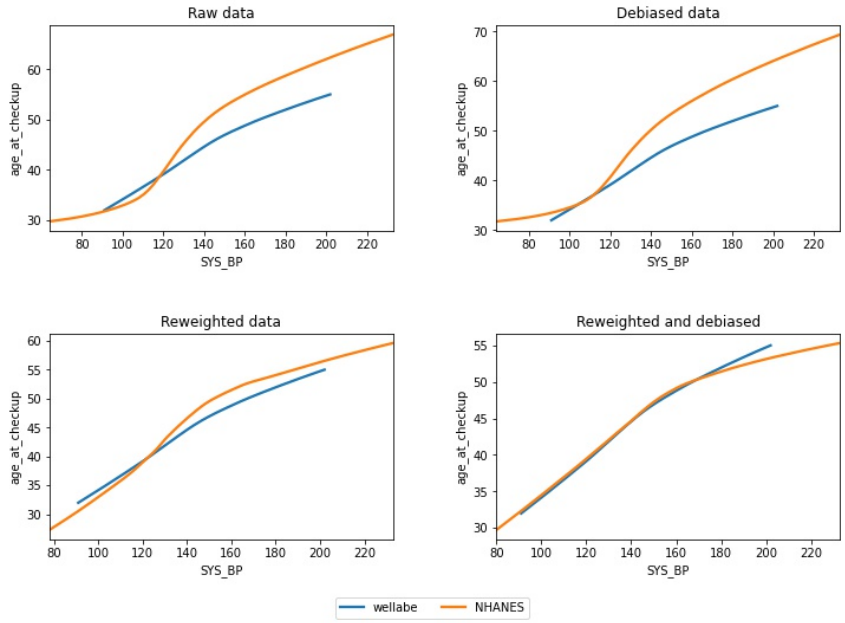

Figure 8: Concept shift and effect of reweighting

Initially, it is clear that NHANES and wellabe do not preserve the same relationship between systolic blood pressure (SYS_BP) and age. Even though de-biasing the data showed better results, reweighting had the best performance. The combined approach gives slight improvement over the reweighting approach, but discards a lot of data, so we generally chose to use the reweighted version of NHANES.

We haven't found a direct method to evaluate the concept shift, so we suggest a proxy that can help to evaluate the concept shift for categorical target variable y. The idea is simple:

1. Select features x to be tested for the concept shift
2. Train two classifiers, f_1 and f_2 on train and test data respectively
3. Predict $\hat{y}_{tr,f_1} = f_1(x_{tr})$ and $\hat{y}_{te,f_1} = f_1(x_{te})$

4. Predict $\hat{y}_{tr,f_2} = f_2(x_{tr})$ and $\hat{y}_{te,f_2} = f_2(x_{te})$

5. Compute percentage of instances assigned to the same class by different classifiers:

$$d_{tr} = \frac{1}{|tr|}|\hat{y}_{tr,f_1} - \hat{y}_{tr,f_2}|$$

$$d_{te} = \frac{1}{|te|}|\hat{y}_{te,f_1} - \hat{y}_{te,f_2}|$$

If the accuracy is close to 1, it means that the concept shift is unlikely in this situation. E.g. when performing such evaluation for diabetes with features DIA_BP, CREA, HDL, CHOLESTEROL the accuracy is 0.9999, so no concept shift is detected.

This approach can be adapted to continuous targets as well. The only difference is that instead of computing the number of matching predictions, we need to use a different metric. In case of age, we evaluate the mean absolute distance.

Using this approach, we have experimented with the pair SYS_BP and age_at_checkup. The results can be seen in Table 2. Also we can observe that the suggested method for measuring the concept shift is quite informative for measuring the concept shift. When only one covariate is present one might simply use the correlation coefficient, however this method becomes in handy when there is more than one covariate.

NHANES version	Mean Absolute Difference on NHANES	Mean Absolute Difference on wellabe
Raw	1.73	1.89
Debiased	1.14	0.63
Reweighted	0.23	0.23
Debiased and Reweighted	0.16	0.18

Table 2: Mean Absolute differences on NHANES and wellabe, for various dataset versions of NHANES

4.4 Transfer Evaluation

Although we have mentioned a few methods to evaluate if we were able to fix the mentioned types of shifts, in a practical situation we are more interested not just in how similar NHANES is to the synthetic welalbe dataset, but how well the model, trained on NHANES can generalize to the real wellabe dataset, using synthetic data as a proxy for learning the dataset shift and fixing it.

Our main metric for evaluating the transfer quality is the difference between the performance of the model on NHANES hold out test set and real wellabe dataset, with synthetic dataset being our validation set. The model metric itself was chosen based on the task and model as explained in Section 5.1 and Section 6.1

5 Age Prediction

Age prediction was chosen as one of the tasks for evaluating model transfer due to its simplicity and ease for the result interpretation. We performed our experiments in the following way. First, the models were trained and tested on the NHANES dataset and verified on real wellabe. Then we reweighted NHANES using KLIEP, described in Chapter 4, and again tested on real wellabe and reweighted NHANES.

In the case with age prediction, we have both covariate and concept shifts. The concept shift happens when we talk about age versus systolic, diastolic blood pressures as well as heart rate. We reweighted the samples, including the y-variable to only capture the relevant samples, as explained in Section 4.3. This way we fixed the joint distribution $p(x, y)$.

Additionally, we have trained the same models directly on the synthetic dataset to get an idea of the upper bound for the model performance.

The main metric we used for performance evaluation is Mean Absolute Error (MAE). We chose it for its good interpretability - it is easier to reason about such outcome variable as tangible as age because we compare the scores in terms of years.

5.1 Model and Metric Choice

We introduced a set of regression models suited for modeling the age prediction task. We found that the ordinary linear regression model and three linear regression models with regularization, namely, lasso regression, ridge regression, and ElasticNet, performed the best (with results recorded in Section 5.3). We also experimented with support vector regression (SVR), multivariate adaptive regression splines (MARS), generalized additive models (GAM), and extreme gradient boosting (XGBoost), however, these models did not give good model performance and thus were not tested on real wellabe. Since the overall modeling procedure was complex having three datasets involved, we provide a general overview of the workflow used during the course of this project (see Appendix F).

We assumed the "simple" regression models to outperform the others due to their better generalization capabilities on both the unsampled and resampled NHANES datasets. This property lies in the nature of the relatively simple model constructions. Simple linear regression models are fitted using the least squares approach only, whereas lasso regression introduces a L1-norm and ridge regression a L2-norm penalty to the objective function. Both lasso and ridge regressions aim to reduce variance at the cost of bias by penalizing large weights. ElasticNet rather builds on the trade-off between both methods by incorporating both L1-norm and L2-norm penalties. Even though there are some variations in the models construction, the results in Section 5.3 show that the overall performance differences between the four models can be neglected.

5.2 Feature Choice

For multiple regression models, it is crucial to ensure that the features are not multicollinear. Multicollinearity suggests high intercorrelations between two or more independent variables and can lead to misleading results when analyzing each independent variable's ability to predict or explain the corresponding dependent variable. Variance

inflation factor (VIF) measures exactly the amount of multicollinearity that exist in a set of independent variables [18]. VIFs start at 1 and have no upper bound. A VIF value of 1 indicates that there is no correlation between one particular independent variable to any others. A large VIF, with value above 5, on the other hand, indicates high multicollinearity and requires correction.

Within the scope of above-mentioned usable features in Section 3.2.2, we performed VIF testing and removed variables with values higher than 5 during the age prediction task. Additionally, we also removed features that have strong pairwise correlations to each other based on Pearson's correlation coefficient. The remaining features are: AST, CHOLESTEROL, CREA, DIA_BP, GGT, HDL, HEART_RATE, SYS_BP, WAIST_SIZE, has_diabetes, height, is_pregnant, and sex. This set of features gave the overall best performance for the age prediction task and was used to produce the results in Section 5.3.

5.3 Results

The results of the experiments are recorded in Tables 3 and 4. The upper bound, obtained from training and testing the same models on the synthetic dataset has MAE score of 8.33. We can see that we achieved this upper bound on real wellabe with all the models after reweighting in Table 4. We observe that after reweighting, the scores for NHANES and wellabe get closer, meaning that we "simplified" the NHANES dataset and made it easier to explain with our chosen models. Reweighted models reduce average MAE from 8.765 to 8.323, which allows us to conclude that the original model was already close to the true model that explains both NHANES and wellabe.

Model Name	MAE (NHANES)	MAE (wellabe)	MAE (diff)
Linear	11.052	8.304	2.748
Lasso	12.055	8.932	3.123
Ridge	12.070	8.928	3.142
ElasticNet	12.034	8.896	3.138
Average	11.803	8.765	3.038

Table 3: Age prediction results with models trained on the raw NHANES dataset

Model Name	MAE (NHANES)	MAE (wellabe)	MAE (diff)
Linear	8.388	8.346	0.042
Lasso	8.388	8.346	0.042
Ridge	8.347	8.317	0.030
ElasticNet	8.307	8.283	0.025
Average	8.358	8.323	0.034

Table 4: Age prediction results with models trained on the resampled version of NHANES dataset using KLIEP

6 Classification Models

After exploring NHANES, we have discovered several categorical variables that might be of particular interest for us, which include asthma, osteoporosis, liver diseases, and so on. Ideally, we would like to predict a score for the disease which will indicate if the client needs to perform other tests. The only relevant variable that was present both in NHANES and wellabe was diabetes. As we have already discussed in Section 3.2.2, glucose level was not one of the usable features due to its high percentage of missing values (51.5% missing after cleaning). Therefore, we were left with the challenge to predict diabetes without the most important variable. In order to have more variety of tasks and a better idea of transfer quality, we have also introduced several artificial categorical variables by discretizing existing continuous variables, which include alanine transaminase (ALT), cholesterol, and BMI (see Section 6.3). The results for each of the classification tasks are discussed in Sections 6.2 and 6.3.

6.1 Model and Metric Choice

From the set of selected models we found, that ordinary logistic regression, Naive Bayes, and Support Vector Machines (SVM) performed the best on the introduced datasets (see Section 6.2). We also looked at different k-nearest neighbour (kNN) models and Decision Trees, none of which improved our results. We make use of the same workflow introduced in Section 5.1 and shown in the Appendix F using the aforementioned classification models instead.

Recall ($\frac{TP}{TP+FN}$) was chosen as our primary evaluation metric, since the goal of health-related feature classification is to capture as many true positives (TP) as possible and minimize false negatives (FN). In the case of multi-class classifications, micro-averaged recall scores were used. Macro-average computes the recall score independently for each class and then takes the average afterwards, meaning all classes are weighted equally. Micro-average, on the other hand, aggregates the contributions across all classes to compute the average recall, and is favored when there exists class imbalance [19].

The best performing models were logistic regression, SVM, and Naive Bayes. Whereas logistic regression and SVM are discriminative models, conditioned by their predictor variables, Naive Bayes is a generative model, aiming to estimate the underlying joint distribution of its predictors. Unlike conditional models, logistic regression and SVM, Naive Bayes is known to be better suited for handling imbalanced datasets.

6.2 Diabetes Prediction

When modeling for diabetes we chose three models: logistic regression as a simple baseline, Naive Bayes due to its good handling of rare events and SVM for its flexibility. We have evaluated all three models with all the reweighted methods we have introduced in Sections 4.1. The results are shown in Table 6. We can observe that none of the methods are better in all cases and for all models, but all of them improve the recall score while still maintaining reasonable accuracy.

Similar to the age prediction experiment, we also trained and evaluated these models on the synthetic dataset to get the idea of the best achievable score, which can be seen

in the column "wellabe" of Table 6.

As we have mentioned, our main metric for evaluation of diabetes is recall, but at the same time we still want our classifier to have an acceptable accuracy. The best trade-off between accuracy and recall was achieved by Naive Bayes model with an accuracy of 0.64 and a recall of 0.27, while on raw NHANES this classifier had a recall of 0.03.

When modeling diabetes, we faced several issues which complicated the overall classification task. First, the distributions of people having diabetes in NHANES and synthetic wellabe dataset are very different which can become problematic for generative models like Naive Bayes as explained in Section 4.2.

Second, we apply different reweighting methods on the NHANES dataset which all rely on learning the distributions from the synthetic dataset. Due to the limitations of the synthetic dataset (Section 2.3), the reweighted dataset mirrors the underlying data generating process of the synthetic dataset, which might be different from the real. We conclude that this is a general issue for rare events, since every positive sample has a lot of weight in modeling and after reweighting we might make this data even more sparse.

Model	Reweighting	Accuracy				Recall			
		nhanes	real wellabe	wellabe	nhanes diff	nhanes	real wellabe	wellabe	nhanes diff
LogReg	Raw	0.74	0.91	0.54	0.17	0.81	0.04	0.56	0.78
	NN	0.7	0.68		0.02	0.82	0.18		0.64
	LogReg	0.83	0.84		0.01	0.91	0.14		0.77
	KLIEP	0.75	0.88		0.13	0.8	0.09		0.71
	BDT	0.73	0.71		0.02	0.75	0.1		0.65
SVM	Raw	0.79	0.89	0.68	0.09	0.63	0.08	0.33	0.55
	NN	0.74	0.73		0.03	0.81	0.13		0.68
	LogReg	0.9	0.83		0.01	0.81	0.11		0.7
	KLIEP	0.75	0.76		0.06	0.85	0.13		0.72
	BDT	0.73	0.71		0.04	0.83	0.16		0.67
NB	Raw	0.86	0.94	0.98	0.09	0.35	0.03	0.01	0.32
	NN	0.72	0.63		0.09	0.72	0.25		0.47
	LogReg	0.83	0.79		0.04	0.8	0.1		0.7
	KLIEP	0.54	0.5		0.04	0.83	0.38		0.45
	BDT	0.61	0.64		0.03	0.8	0.27		0.53

Table 5: Diabetes prediction results with models trained on the resampled version of NHANES dataset using different resampling methods.

Third, we found the glucose level biomarker to be the one most important predictor for a person to have diabetes (Appendix E). Due to the high sparsity of this feature over all year-buckets in NHANES we had to drop it and thus could not use it for modeling (Section 3.2).

Lastly, the nature of the target variable itself is problematic. The presence of diabetes is established based on questionnaires, so it only reflects the cases when the disease is diagnosed and the respondent knows about it. Therefore, it is inherently hard to find a good classifier, since the ground truth labels contain false negatives (undiagnosed diabetes).

We conclude that reweighting NHANES improves the performance for all models. However, the recall scores are generally low, even when trained directly on synthetic

dataset. so even if we find the region that is closest in feature space to wellabe, the data might still not be close enough or even not good enough for explaining the target variable.

6.3 Discretized Features

6.3.1 BMI, ALT, and CHOLESTEROL Predictions

In order to create classes in features, discretization of features was necessary. This was done in accordance with the normal healthy ranges, as discussed in Section 3. For example, if the target variable was HEART_RATE, patients below the normal range were given a value 0, if they were within the range, the value given was 1 and above the range was 2. The goal was to see if we could train a model on NHANES that could predict whether a patient in wellabe had a normal, low or high resting heart rate.

The chosen features to discretize were: BMI, CHOLESTEROL and ALT. We chose these features in order to get a well-rounded perspective of all the overlapping features between NHANES and wellabe. It is important to note that all features mentioned above lead to more balanced classes, which gives better starting conditions than diabetes. The features were also chosen in accordance to the number of years that the biomarker was collected in NHANES; the more years, the better (see Appendix A). For this reason, spirometry data was not considered.

BMI was chosen to represent all the features that are calculated from other biomarkers. This served as a baseline prediction to see whether a directly correlated feature can be discretized and predicted; it will not only serve as an evaluation for model transfer, but as an assessment of discretization in medical data.

Target	Reweighting	Accuracy and Recall			
		nhanes	real wellabe	wellabe	nhanes diff
BMI	Raw	0.89	**0.79**	0.7	0.1
	NN	0.85	**0.79**		0.06
	LogReg	0.99	0.64		0.35
	KLIEP	0.86	0.78		0.08
	BDT	0.88	0.75		0.13
ALT	Raw	0.93	0.86	0.85	0.07
	NN	0.88	**0.87**		0.01
	LogReg	0.99	0.63		0.36
	KLIEP	0.96	0.83		0.13
	BDT	0.86	**0.86**		0.0
CHOL	Raw	0.59	0.67	0.72	0.08
	NN	0.71	**0.71**		0.0
	LogReg	0.98	0.69		0.29
	KLIEP	0.69	**0.71**		0.02
	BDT	0.54	0.67		0.13

Table 6: Discretized BMI, ALT, and CHOLESTEROL prediction results with logistic regression models using different reweighting methods for the NHANES dataset.

Next, CHOLESTEROL shows perspective on biomarkers that fluctuate often, in the sense that a change in diet or stress can result in a change of cholesterol levels in as

short as a couple of weeks. This feature is very important to see how other biomarkers change in a shorter timespan and whether there exists a minimal set that can predict this fluctuation. Moreover, a high cholesterol value is strongly correlated to stress, a problem faced often in wellabe's demographic - the corporate space.

Finally, ALT was chosen as the biomarker that can detect organ problems. ALT is an enzyme that breaks down proteins in the liver for example, and its release into the bloodstream indicates organ problems. Predicting this biomarker acts as an example of getting indications in a patient regarding a very specific illness or disease (in our case organ failure). Many of the features in the metabolism group follow the same structure as ALT and accurately represented the majority.

6.3.2 Results

In Figure 6, the results for the discretized model predictions are shown. The table shows the target variables, as described above, and the accuracy/recall scores for each of the reweighting methods applied. In our case of micro-averaging, the accuracy and recall scores have the same value. The models were trained on the NHANES dataset with the reweighting methods described in Chapter 4, and tested on both NHANES and the real wellabe dataset - the differences between these two scores was also recorded. Additionally, the column labelled "wellabe" is the result for the logistic regression model that was directly trained and tested on the synthetic wellabe dataset. It therefore serves as our benachmark for model performance.

It is important to note that in the case of reweighting with LogReg (see Section 4.1.2.2), the accuracy and recall tested on NHANES increases to almost perfect, while the accuracy and recall tested on wellabe drastically decreases. In this case, we have resampled NHANES and trained to overfit the model, which explained less of the wellabe dataset. This is because the resampling method of dataset prediction made the NHANES variety much smaller, and easier to fit.

When looking at the model predicting the BMI feature, we saw that simply using only the raw NHANES dataset to train performs significantly better on the real wellabe dataset than training on the wellabe dataset. This is an example where the model trained on the NHANES datset, which we consider more global in the sense that it accounts for the true population, is better than training on a local model. Additionally, when we resample the NHANES dataset using NN, KLIEP and BDT (see Section 6), our performance on the real wellabe dataset is still better than our wellabe benchmark. The same argument is made for ALT.

Next, we can see for all of the discretized models, excluding reweighting using LogReg, the difference between the NHANES and wellbabe scores are relatively small. This means that both the model transfer and predicting discretized features is possible when considering these two datasets. Additionally, the fact that reweighting showed minor improvements in comparison to the raw data, indicates that we achieved close to the true model that can predict these discretized features.

Finally, we see for CHOLESTEROL that reweighting with the NN and KLIEP method performs significantly better and similarly on NHANES and wellabe, and almost attains the upper bound that we have set for training and testing on wellabe (0.72).

7 Beyond wellabe: New Labels

To further benefit from the richness in variables that NHANES offers, we decided to include additional labels for classification modeling for two main reasons. Firstly, we can predict further diseases that are recorded in NHANES and make use of additional features which are not available in wellebe. In other words, those variables to include can act both as target variables as well as features for classification. Based on feature importance, if a group of features can explain a certain disease well and is not present in wellabe, we could suggest those features to be included in order to add values to the wellabe dataset in the future. For this purpose, we chose asthma as our target variable (explained in Section 7.1). The other main objective is to identify possible risk-factors that the wellabe samples have, but are not recorded. For this task, we modeled the medication status, specifically if prescribed medication was used within the last 30 days (see Section 7.2). We noticed that wellabe does keep track of sample's medication usage, but only a limited set. Ideally, if the classification model can perform well on NHANES using the features that exist in both wellabe and NHANES datasets, we can reason about how well the model can potentially transfer to wellabe, base on our previous transfer evaluations mentioned in Chapter 6. A good performing model could add insights and recommend wellabe to further check samples who are classified as medication takers, but with no medication history recorded.

To identify the most suitable target variables to model, we included 20 disease or lifestyle-related labels to the NHANES dataset, which have values in at least five year-buckets. The set of additional features we added include, for example, asthma, arthritis, bronchitis, days feeling physically unwell, medicine used during the last 30 days, and number of hours using computer, and so on.

Furthermore, we researched on the medical meanings for each disease label, and mapped each disease to its most relevant biomarkers (see Appendix E). We used the same sources and procedure described in section 3.1. If there is a medically approved relationship between a feature and and a certain disease, then 1 is labelled in that respective row and column, whereas empty space suggests no significant medical relation. We used this table to make sense of the generated predictors, particularly for asthma prediction.

7.1 Asthma Prediction

For the asthma classification task, the target variable is a binary label containing information if a person has ever been diagnosed with asthma by doctors. We chose this task not only because there are almost no missing values in NHANES from the years 1999 - 2018, but wellabe has a general interest in this label and already incorporated it into their questionnaires. To include as many respiratory features as possible (their importance in relation to asthma is shown in the mapping under Appendix E), we only used the NHANES data from 2007 until 2012 due to respiratory features' limited availability (shown in Appendix A). Furthermore, we chose and imputed features with less than 5% missing values in the above-mentioned year range from all the available features we had. In the end, logistic regression gave the best overall model performance with an accuracy score of 0.66 and a recall score of 0.63 on the NHANES testset. Coefficients of this model can be found in Figure G under Appendix G. Based on the coefficients, we observed that in the additionally added labels, only bronchitis and medicine used in last 30 days were

good indicators for identifying asthma history. However, based on the coefficients, the feature-disease mapping was supported. In other words, features like DIA_BP, FEF25, and MAP contributed greatly to identifying asthma in the model.

7.2 Medication Prediction

We found, that wellabe tracks some data about the medication status of their patients within a questionnaire. They ask for specific medications such as anti-diabetes tablets or insulin, antihypertensive drugs, blood lipid lowering drugs (statins), blood thinner, thyroid medication, painkiller, cholesterol lowering drugs or others. This motivated us to predict if a survey participant in NHANES took prescribed medicine in the last 30 days. We propose the resulting model to wellabe, to use it for identification of patients taking any medication which is not so far tracked in the questionnaire. Additionally the model can be used to infer information about a patients current medication status in the future.

Reweighting	Accuracy	Recall
Raw	0.71	0.72
NN	0.64	0.65
LogReg	0.79	0.76
KLIEP	0.69	0.66
BDT	0.66	0.66

Table 7: Prediction whether a person took medication in the last 30 days.

When transfering the model trained on NHANES, we can not directly validate its performance on wellabe due to the lack of a comparable target variable in the wellabe dataset. Therefore, we use the test results of NHANES in combination with the knowledge gained in terms of model transfer from diabetes and the discretized features in Section 6.2 and Section 6.3. To model the classification task, we applied logistic regression, since it performed well on all previous classification tasks. The set of features we use contains ALT, CHOLESTEROL, CREA, DIA_BP, GGT, HDL, HEART_RATE, WAIST_SIZE, sex, age_at_checkup and is_pregnant. Again we make use of the synthetic wellabe datasets joint distribution by reweighting NHANES using NN, LogReg, KLIEP, and BDT.

The results in Table 7 show that the model trained on the raw and reweighted NHANES datasets all have a good accuracy and recall scores and thus are suited for predicting the medication status in the NHANES dataset. From the prediction models in discretized features in Section 6.3, we saw that after resampling, the difference in accuracy and recall scores for NHANES and wellabe decreased. This was true in all of the reweighting methods except for LogReg. From this we saw in the discretized case that a drop in accuracy and recall after resampling, would increase the accuracy and recall on wellabe. We can expect that the reweighting method would perform similarly on the real wellabe dataset for the medication status.

The final evaluation of the model performance on real wellabe can not be covered within the scope of this project and thus is left for future work.

8 Conclusions

The core focus of this work was to investigate the possibility and limitations of transferring a model from one dataset to another. The key problem that we were facing was the dataset shift. Although both NHANES and wellabe are subsets of the global human population, finding a true model that will explain the whole population is not always possible. Hence, we had to address this problem using methods, explained in Chapter 4.

Our main conclusion is that even though reweighting methods can be quite effective as we demonstrated in Section 4.1.3, this is not a universal tool which can be applied for all problems. We see that if the concept is generally learnable from the synthetic data (which we estimated by training and evaluating on the synthetic dataset), then the model will transfer well. However, if the concept is not explainable by the subset of population that we have, reweighting might give only slight improvement or sometimes even make the performance worse. This is especially problematic with rare events that are not well represented in the data, as discussed is Section 6.2.

We also discovered that sometimes simple, naive methods can work surprisingly well. Even though we found and tried different sophisticated methods, the nearest neighbour approach suggested in Section 4.1.2.1, provided the best improvement for learnable concepts like discretized BMI, CHOLESTEROL, and ALT.

In the case of hard concepts, like diabetes, NN significantly improves the trade-off between accuracy and recall. After resampling, we were able to detect 18% of the people who actually have diabetes instead of 4%, while still providing reasonable accuracy.

However, NN is very computationally expensive due to the necessity of computing all pairwise distances between the two datasets, as shown in Section 4.1.2.1. Therefore, we also offered a variety of second best methods. There is no specific second best winner, but KLIEP and BDT both provided rather good results, with BDT being significantly faster than NN. Hence, we would suggest BDT as a second best choice when resampling when NN is not feasible.

Results for the dataset prediction method are rather contradictory. Although on a hard task it provides reasonable improvement, for easier tasks it focuses the model too much on the part of the NHANES dataset that matches the synthetic dataset, which results in significant improvement of accuracy on NHANES but poor results on wellabe.

The scarcity of medical data and the importance of medical accuracy makes this an ongoing challenge where the quality of patient diagnosis is consistently being improved using machine learning. Whether it is incorporating more data from rich datasets such as NHANES, or trying to expand to a different demographic region, we have shown that being able to transfer a model from a different source with the provided methods of this project is possible.

Bibliography

[1] Wellabe.de. URL: https://www.wellabe.de/en/en/.

[2] Jose Moreno-Torres et al. "A unifying view on dataset shift in classification". In: *Pattern Recognition* 45 (Jan. 2012), pp. 521–530. DOI: 10.1016/j.patcog.2011.06.019.

[3] *About the National Health and Nutrition Examination Survey*. Sept. 2017. URL: https://www.cdc.gov/nchs/nhanes/about_nhanes.htm.

[4] Harvard Health Publishing. *Health Information and Medical Information*. URL: https://www.health.harvard.edu/.

[5] *Health Library*. URL: https://my.clevelandclinic.org/health.

[6] *CDC A-Z Index - H*. URL: https://www.cdc.gov/az/h.html.

[7] Donald Rubin. "Inference and missing data". In: *Biometrika* 63.3 (Dec. 1976), pp. 581–592. ISSN: 0006-3444. DOI: 10.1093/biomet/63.3.581.

[8] *Facts About Hypertension*. Sept. 2020. URL: https://www.cdc.gov/bloodpressure/facts.htm.

[9] Amos Storkey. "When Training and Test Sets Are Different: Characterizing Learning Transfer". In: *Dataset Shift in Machine Learning* (Jan. 2009), pp. 3–28. DOI: 10.7551/mitpress/9780262170055.003.0001.

[10] J. L. Hodges. "The significance probability of the smirnov two-sample test". In: *Arkiv för Matematik* 3 (1958), pp. 469–486.

[11] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of Statistical Planning and Inference* 90 (Oct. 2000), pp. 227–244. DOI: 10.1016/S0378-3758(00)00115-4.

[12] Steffen Bickel, Michael Brückner, and Tobias Scheffer. "Discriminative learning for differing training and test distributions". In: *ACM International Conference Proceeding Series* 227 (Jan. 2007), pp. 81–88. DOI: 10.1145/1273496.1273507.

[13] Masashi Sugiyama et al. "Direct importance estimation for covariate shift adaptation". In: *Annals of the Institute of Statistical Mathematics* 60 (Feb. 2008), pp. 699–746. DOI: 10.1007/s10463-008-0197-x.

[14] Alex Rogozhnikov. "Reweighting with Boosted Decision Trees". In: *Journal of Physics: Conference Series* 762 (Aug. 2016). DOI: 10.1088/1742-6596/762/1/012036.

[15] Marco Saerens, Patrice Latinne, and Christine Decaestecker. "Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure". In: *Neural computation* 14 (Feb. 2002), pp. 21–41. DOI: 10.1162/089976602753284446.

[16] John Elflein. *Adult prevalence of diabetes in selected countries 2019*. Jan. 2020. URL: https://www.statista.com/statistics/236764/prevalence-of-diabetes-in-selected-countries/.

[17] Alexey Tsymbal. "The Problem of Concept Drift: Definitions and Related Work". In: *Computer Science Department, Trinity College Dublin* 106 (May 2004), p. 58.

[18] Robert M. O'brien. "A Caution Regarding Rules of Thumb for Variance Inflation Factors". In: *Quality & Quantity* 41.5 (Oct. 2007), pp. 673–690. ISSN: 1573-7845. DOI: 10.1007/s11135-006-9018-6. URL: https://doi.org/10.1007/s11135-006-9018-6.

[19] Marina Sokolova and Guy Lapalme. "A systematic analysis of performance measures for classification tasks". In: *Information Processing & Management* 45.4 (2009), pp. 427–437. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2009.03.002.

Appendices

A Feature Availability in NHANES

Year	Body Measurements	Biochemistry Profile	Body Impedance	ABI	Spirometry	DEXA
1999-2000	✓	✓	✓	✓		
2001-2002	✓	✓	✓	✓		
2003-2004	✓	✓	✓	✓		
2005-2006	✓	✓				
2007-2008	✓	✓			✓	
2009-2010	✓	✓			✓	
2011-2012	✓	✓			✓	✓
2013-2014	✓	✓				✓
2015-2016	✓	✓				✓
2017-2018	✓	✓ (except TRIGLY)				✓
	SYS_BP, DIA_BP, HEART_RATE, WEIGHT, WAIST_SIZE, height	GLUCOSE_LEVEL, TRIGLYCERIDES, CHOLESTEROL, HDL, AST, ALT, GGT, CREA	BODY_WATER, BODY_FAT, FAT_KG	ABI_L, ABI_R	FVC_L, FEV1_L, PEF_L, FEF25	BONE_MINERAL_MASS

Table 8: Availability of the feature groups in NHANES across all years

B Feature Description and Normal Ranges

Feature	Description	Normal Range (Male)	Normal Range (Female)
SYS_BP	Systolic Blood Pressure: The amount of pressure that the heart puts on the arteries with each beat.	90-120 mm Hg	90-120 mm Hg
DIA_BP	Diastolic Blood Pressure: The pressure in the arteries in between each heartbeat	60-80 mm Hg	60-80 mm Hg
PP	Pulse Pressure: The difference between the Systolic Blood Pressure and Diastolic Blood Pressure $PP = SYS\_BP - DIA\_BP$	40-60 mm Hg	40-60 mm Hg
MAP	Mean Arterial Pressure: The average arterial pressure $MAP = \frac{SYS\_BP + 2(DIA\_BP)}{3}$	70-100 mm Hg	70-100 mm Hg
ABI	Ankle-Brachial Index: The ratio of blood pressure in the arm and the ankle	1.0-1.5	1.0-1.5
HEART_RATE	Resting Heart Rate: Number of heart beats in 60 seconds while the patient is at rest.	60-100 bpm	60-100 bpm
GLUCOSE_LEVEL	Glucose Level: Measures the blood sugar level.	70-125 mg/dL	70-125 mg/dL
CHOLESTEROL	Cholesterol: Found in the body tissue, it is used to build cells, produce hormones and vitamins $CHOLESTEROL = HDL + LDL + 0.2(TRIGYCERIDES)$	100-200 mg/dL	100-200 mg/dL
TRIGLYCERIDES	Triglycerides: Fat (lipid) level in the blood.	<150 mg/dL	<150 mg/dL
HDL	High-Density lipoproteins: Responsible for removing excess cholesterol in the blood.	>40 mg/dL	>50 mg/dL
ALT	Alanine Aminotransaminase: An enzyme used in the liver to break down proteins. It is often released into the blood when liver problems occur	7-40 U/L	7-35 U/L
AST	Aspartate Aminotransaminase: Similar to ALT, it is an enzyme found in the liver, heart, kidneys, lungs and various muscles	8-48 U/L	8-48 U/L
GGT	Gamma-Glutamyl Transferase: Similar to ALT, it is an enzyme common in the liver and other organs.	8-61 U/L	5-36 U/L
CREA	Creatinine: An amino acid stored in muscles used for energy	0.8-1.3 mg/dL	0.6-1.1 mg/dL

Feature	Description	Normal Range (Male)	Normal Range (Female)
FLI	Fatty Liver Index: Indicator of fat levels in the liver $Constant$=0.953($TRIGLYCERIDES$)+0.139(BMI) +0.718($log_e GGT$)+0.053($WAIST\_SIZE$)−15.745 $FLI = \frac{e^{Constant}}{1+e^{Constant}}$	<30	<30
OXYGEN_SATURATION	Oxygen Saturation: The percentage of oxygen that circulates through the body	95-100 %	95-100 %
FVC_L	Forced Vital Capacity: The maximum amount of air that can be exhaled.	$\frac{FEV1\_L}{FVC} > 0.7$	$\frac{FEV1\_L}{FVC} > 0.7$
FEV1_L	Forced Expiratory Volume in 1 Second: The amount of air that can be exahled in one second.	$\frac{FEV1\_L}{FVC} > 0.7$	$\frac{FEV1\_L}{FVC} > 0.7$
PEF_%	Peak Expiratory Flow as a Percentage: The expiratory flow percentage of the maximum achieved expiratory flow .	80% or higher	80% or higher
PEF_L	Peak Expiratory Flow in Litres: Maximum expiratory flow.	7.5-11.5 L/s	5-8 L/s
FEF25	Forced Expiratory Flow at 25% of the Lung Volume: The flow of exhaled air after 25% has already been exhaled.	50-60% and higher of the predicted value based on age and height	50-60% and higher of the predicted value based on age and height
height	Body Height: Measured in centimeters	Depending on age	Depending on age
WEIGHT	Body Weight: Measured in kilograms	Depending on HEIGHT	Depending on HEIGHT
VISCERAL_FAT	Visceral Fat Level: The amount of fat near the organs in the abdomen	Depending on waist size	Depending on waist size
DAILY_CALORIC_NEEDS	Daily Caloric Needs: Recommended number of calories needed per day	2000-3000	1600-2400
MUSCLE_MASS	Muscle Mass: Percentage of weight that is muscle	31-44%	26-33%
FAT_KG	Body Fat Mass: The weight of fat in the patient	Depending on BODY_FAT	Depending on BODY_FAT
BODY_FAT	Body Fat Percentage: The percentage of fat in the body	8-25% depending on age	14-32% depending on age
BODY_WATER	Body Water Percentage: Percentage of water in the body $BODY\_FAT = \frac{FAT\_KG}{WEIGHT} \times 100$	43-73%	41-63%
BODY_MINERAL_MASS	Bone Mineral Mass: Amount of minerals contained in the bone	2.5-3.2 kg	1.8-2.5 kg
HIP_SIZE	Hip Size: Measured in centimeters	Depending on the hip size	Depending on the hip size
WAIST_SIZE	Waist Size: Measured in centimeters	Depending on the waist size	Depending on the waist size

Feature	Description	Normal Range (Male)	Normal Range (Female)
BMI	Body Mass Index: Indicator of Body Fatness $BMI = \frac{WEIGHT}{(HEIGHT/100)^2}$	18.5-24.9	18.5-24.9

Table 9: Feature names, descriptions, and their normal ranges

C wellabe and NHANES Features Availability Comparison

Feature Group	Wellabe Feature	Feature Label	Available in NHANES?
Metabolism	Blood Sugar	GLUCOSE_LEVEL	✓
	Cholesterol	CHOLESTEROL	✓
	Triglycerides	TRIGLYCERIDES	✓
	High-Density Lipoproteins	HDL	✓
	Alanine Aminotransaminase	ALT	✓
	Aspartate Aminotransaminase	AST	✓
	Gamma-Glutamyl Transferase	GGT	✓
	Creatinine	CREA	✓
	Fatty-Liver Index	FLI	✓
Cardiovascular System	Systolic Blood Pressure	SYS_BP	✓
	Diastolic Blood Pressure	DIA_BP	✓
	Pulse Pressure	PP	✗
	Mean Arterial Pressure	MAP	✗
	Ankle-Brachial Index	ABI	✓
	Resting Heart Rate	HEART_RATE	✓
Respiratory System	Oxygen Saturation	OXYGEN_SATURATION	✗
	Forced Vital Capacity	FVC_L	✓
	Forced Expiratory Volume in 1 Second	FEV1_L	✓
	Peak Expiratory Flow as a Percentage	PEF_%	✗
	Peak Expiratory Flow in Liters	PEF_L	✓
	Forced Expiratory Flow at 25%	FEF25	✓
Body Composition	Body Height	HEIGHT	✓
	Body Weight	WEIGHT	✓
	Visceral Fat Level	VISCERAL_FAT	✗
	Daily Caloric Needs	DAILY_CALORIC_NEEDS	✗
	Muscle Mass	MUSCLE_MASS	✗
	Body Fat Mass	FAT_KG	✓
	Body Fat Percentage	BODY_FAT	✓
	Body Water Percentage	BODY_WATER	✓
	Bone Mineral Mass	BODY_MINERAL_MASS	✓
	Hip Size	HIP_SIZE	✗
	Waist Size	WAIST_SIZE	✓
	Body Mass Index	BMI	✓
Various	Age at Check-up	age_at_checkup	✓
	Diabetes	has_diabetes	✓
	Health Score	health_score	✗
	Recent Excercise	excercised_recently	✗
	Review Score	review_score	✗
	Pregnancy	is_pregnant	✓
	Recheck	recheck	✗

Table 10: List of features in the wellabe datasets and their availability in NHANES

D Cleaning Rules

Feature	NHANES	wellabe	Cleaning Procedure
CHOLESTEROL	✓	✓	Remove samples with value < 60
TRIGLYCERIDES	✓		Remove samples with values > 3165
GGT	✓		Remove samples with value > 1000
CREA	✓		Remove samples with value > 10
CREA		✓	Convert values < 0.1 to NaN
BMI	✓		Remove samples with value > 62.87 (more than 5 SDs from the mean)
BMI		✓	Remove samples with value < 12
DIA_BP	✓		Set values < 1 to NaN
HDL		✓	Convert values < 10 or $= 124$ to NaN
ALT		✓	Remove samples with value < 4
AST		✓	Remove samples with value < 7
WEIGHT		✓	Convert values < 20 to NaN
WAIST_SIZE		✓	Convert values < 50 to NaN
ABI_R, ABI_L		✓	Convert values < 0.12 to NaN
HEART_RATE		✓	Convert values < 30 to NaN
PP		✓	Remove samples with value < 6
FVC_L		✓	Remove samples with value < 1
FEV1_L		✓	Remove samples with value < 0.75
FEF25		✓	Remove samples with value < 0.15
PEF		✓	Remove samples with value < 0.9
FEF25, PEF		✓	For samples with FEF25 $>$ PEF, set FEF25 to 40% of PEF

Table 11: Data cleaning rules for both NHANES and wellabe datasets

E Feature-Disease Mapping

Feature	Asthma	Arthritis	Bronchitis	Congestive Heart Failure	Coronary Disease	Diabetes	Emphysema	Kidney Disease	Liver Disease	Osteoporosis	Thyroid Disease
SYS_BP	1			1	1	1					1
DIA_BP	1			1	1						
PP	1			1	1						1
MAP	1			1				1			
ABI				1	1						
HEART_RATE				1							1
GLUCOSE_LEVEL						1					1
CHOLESTEROL					1	1					
TRIGLYCERIDES					1	1		1			1
HDL					1						
ALT				1		1			1		
AST									1		
GGT				1		1			1		
CREA					1			1	1		
FLI						1			1		
FVC_L	1		1				1				
FEV1_L	1		1				1				
PEF_L	1		1				1				
FEF25	1		1				1				
height		1			1					1	
WEIGHT	1	1			1	1		1	1	1	
BODY_FAT	1				1	1					
BODY_WATER								1			
BODY_MINERAL_MASS										1	
WAIST_SIZE					1	1					

Table 12: Mapping of the features in NHANES to newly identified disease target variables

F Modeling Flowchart

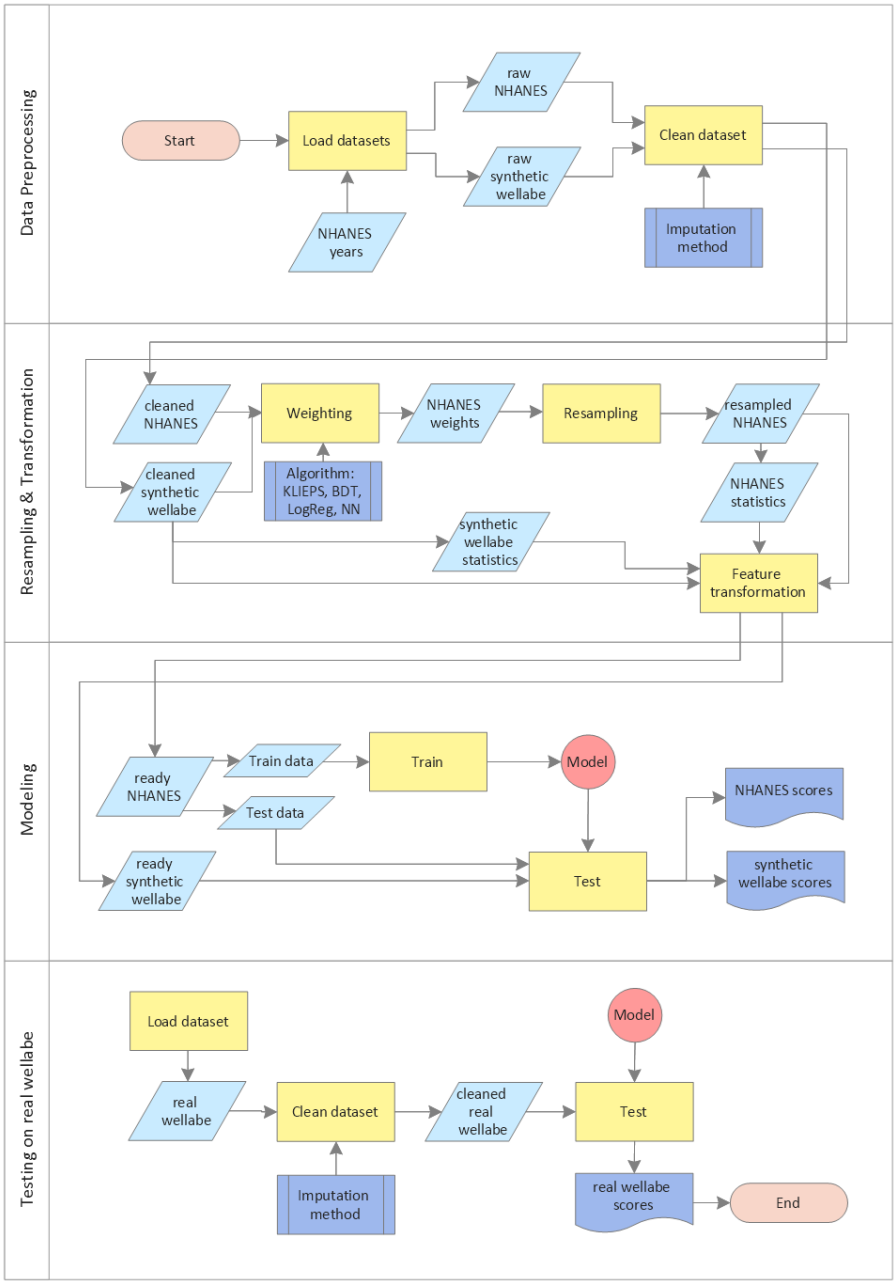

Figure 9: Flowchart of the general modeling procedure used for this project

G Asthma Classification

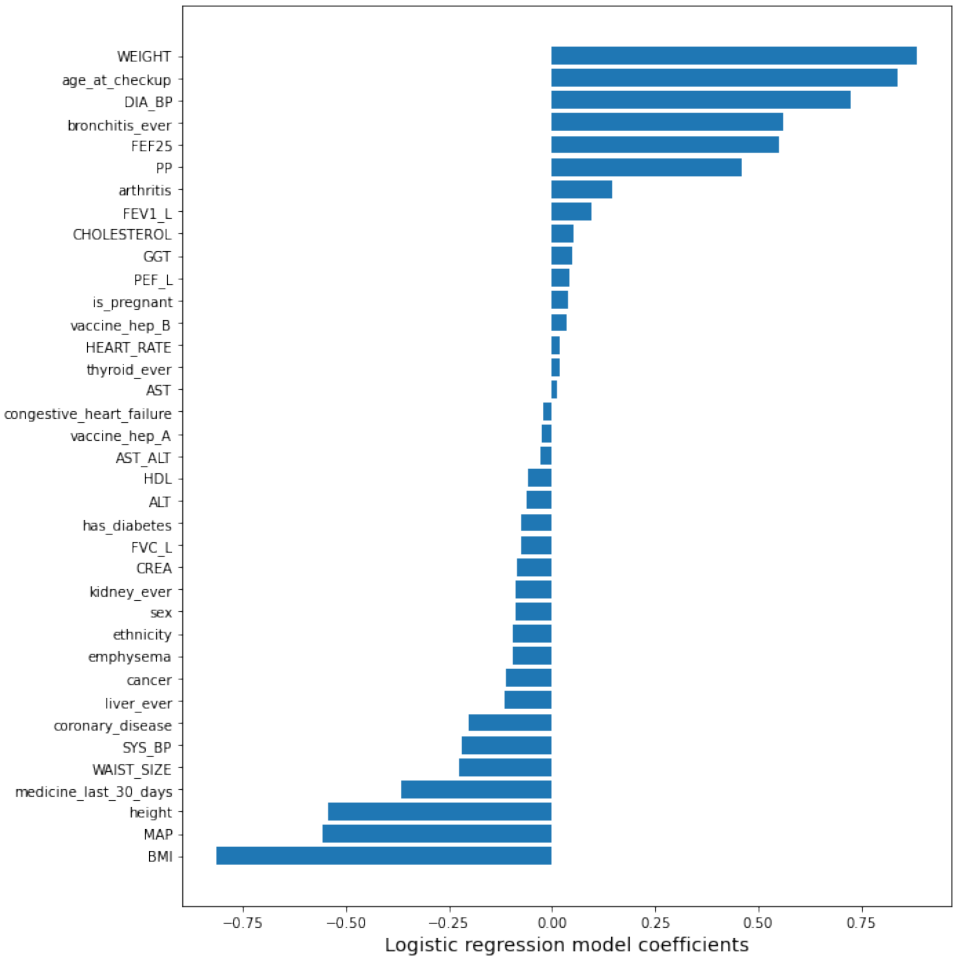

Figure 10: Logistic regression model coefficients and features used for asthma classification