UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Exposome Data Drift: Implications for Machine Learning Based Diabetes Prediction

Author:
Peter Hannagan BROSTEN

Supervisor:
Dr. Karim LEKADIR
Marina CAMACHO

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2023

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Exposome Data Drift: Implications for Machine Learning Based Diabetes Prediction

by Peter Hannagan BROSTEN

Data drift is a problem in machine learning (ML) where characteristics of the input predictors changes over time, leading to model degradation. However, the effects of data drift on ML models built from human exposome data have not been well described yet. This study aimed to investigate data drifts for exposome data in ML models of diabetes risk. 7,521 participants with a diagnosis of diabetes from the UK Biobank, along with a proportional control group from 2006 to 2010 were used to train several baseline ML models for diabetes prediction. A second cohort of 4,007 participants attending the follow-up assessment period from 2012 to 2013 was used to assess potential data drifts over time. When evaluated on the second cohort, significant performance degradation was found in all baseline models (i.e. average precision dropped by 15%, f1-score by 12%, recall by 15%, and precision by 10%). A suite of drift detection tests were run on the best performing baseline models to identify possible signatures of three distinct kinds of data drift: covariate drift, label drift, and concept drift. Utilizing both multivariate and univariate data-distribution based detection methods, covariate drift was identified in features such as Birth Year, BMI, Frequency of Tiredness, and Lack of Education. A comparison of prevalence rates for time-ordered batches of the population found no severe label drift. Nonetheless, gradual label drift could not be ruled out. A model-aware concept drift detection method was employed, monitoring temporal changes in normalized Shapley contributions for the model's input features. This test found drift in abnormal changes in feature contribution when predicting on the second cohort for the Birth Year feature and near alerts in multiple others. This study shows the potential for data drift acting as a driver of model degradation in exposome-based ML models and highlights the need for further research into the traceability of clinical AI/ML solutions.

Acknowledgements

This thesis represents the culmination of a year long effort. This was a journey which could not have happened without the help and support of many along the way.

I would like to thank the European Union's Horizon 2020 research and innovation program which funded this research under Grant Agreement Nº 848158 (early-cause.europescience.eu) and the Grant Agreement Nº 874739 (www.longitools.org); the good people at the BCN-AIM research group, for welcoming and helping me over this last year; and my supervisors Karim Lekadir and Marina Camacho, your wisdom and counsel has been invaluable during this project.

To my parents and sister, thank you for always finding the time to call despite the inconvenience of living on opposite sides of the world. To my friends, Joe and Jessamyn, thank you for listening to me ramble and finding new ways to make me laugh.

Lastly, to my life partner Elena. Your love is what brought me here and is what helped me preserver. You supported me when I was low and cheered me on when I felt on top of the world. I can never put into words what you mean to me. Thank you for everything you are.

Contents

Chapter 1

Introduction

The field of healthcare has witnessed a revolution in recent years with the application of machine learning (ML) techniques, offering immense potential for disease prediction and improving patient care. Among the numerous applications, disease prediction has garnered significant attention due to the transformative impact that early detection and preventive measures can have on long-term prognoses (Uddin et al., 2019). One particular disease that has been a focus for AI/ML predictive models is diabetes. Diabetes is recognized as a major global health crisis with increasing prevalence (Saeedi et al., 2019). A substantial proportion of diabetes cases remains undiagnosed, particularly in the Global South. Early detection of diabetes is crucial for initiating patient-centered management strategies to enhance glycemic control and minimize complications (Chatterjee, Khunti, and Davies, 2017), thereby highlighting the need for high-performing predictive models.

ML models developed for diabetes prediction have demonstrated remarkable accuracy, enabling early detection, timely intervention, and personalized treatment (Jaiswal, Negi, and Pal, 2021). However, a common challenge encountered by ML solutions is the lack of trust from stakeholders. AI/ML solutions are often perceived as complex, opaque, and difficult to comprehend, utilize, and trust in critical clinical applications. Efforts have been made in recent years to address concerns surrounding clinical AI/ML models. Developmental and transition guidelines have been established to ensure the trustworthy and ethical development and integration of such tools (Lekadir et al., 2021; Char, Abràmoff, and Feudtner, 2020; McCradden et al., 2022). These guidelines emphasize principles of best practice to foster complete stakeholder trust, including fairness, universality, and traceability. Fairness aims to mitigate discriminatory group and individual bias, as well as prevent the reinforcement or amplification of existing disparities. Universality encourages standardization to ensure compatibility across diverse settings. Traceability necessitates transparency in model design, including proper documentation of data collection and utilization throughout the model's lifecycle, along with regular monitoring for any potential model degradation.

One crucial assumption made during the AI/ML learning process is that the training data adequately represents both current and future relationships. However, if the training data is derived from non-stationary distributions, there is a risk of rapid model degradation as predictions are made further into the future (Hoens, Polikar, and Chawla, 2012). This phenomenon is referred to as data drift, where the statistical properties or relationships of the input data change over time, leading to a mismatch between the training and future data distributions. Even though all sources of data risk data drift, some may be less robust against it. Precisely the exposome, as it spans a wealth of information from environmental exposures to lifestyle and dietary choices, may experience more temporal drift reflecting changes in cultural norms and emerging technologies. There has been limited investigation

conducted regarding the stationarity of exposome data nor the temporal impact on exposome-based model performance. For exposome data, data drift can manifest in various forms, such as shifts in patient demographics, changes in healthcare policies, advancements in medical technologies, or even seasonal variations. Regardless of the source, data drift poses a significant threat to the performance and reliability of ML models, potentially resulting in erroneous predictions and compromised patient outcomes.

Understanding exposome-based ML models' susceptibility to data drift is crucial for ensuring their effectiveness and maintaining initial predictive power over time. In this thesis, various exposome-based diabetes risk-predictive baseline models were trained and evaluated using the UK Biobank data set. The performance on two evaluations sets were analyzed, the first coming from the same time period but consisting of participants assessed at locations the training data omitted and the second consisting of assessments conducted between two and three years after the final training assessment. A significant drop in performance was found in all baseline models between the former and the latter. Multiple types of analysis were conducted to identify the drivers of this model degradation including multivariate and univariate drift detection methods (covariate drift), a performance-aware detection method using Shapley feature importance analysis (concept drift), and an analysis of time-ordered diagnosed prevalence rates of diabetes within the dataset (label drift). While no significant label drift was found, possible instances of covariate and concept drift were identified.

All code used for data prepossessing, cohort selection, model development, and drift analysis is provided here. Due to grant agreements, the data used is only available via application to UK Biobank.

Chapter 2

Preliminaries

The information in this section forms a foundation upon which this thesis was built. The primary focus of this section is predominantly Bayesian statistics, probability distribution comparison, and the topic of data drift. Most definitions are drawn from various well regarded texts on these subjects. The reader is expected to have a general understanding of statistics and machine learning. Readers well versed in the subjects of type of data drift and methods of their detection may skip ahead to Chapter 3 and simply refer back to the prior sections when needed.

2.1 Probabilistic Underpinnings

2.1.1 Random Variables and Probability

A *random variable* $X\colon S \to \Omega_X$, is a function from a sample space S to an outcome space Ω_X. If the image of a random variable is countable, like a binary variable where $\Omega_X = \{0, 1\}$, we say X is a *discrete random variable*. In the case that the our outcome space is non-numeric, e.g. $\Omega_X = \{\text{dog}, \text{cat}, \text{bird}\}$, we refer to it as a *categorical random variable*. When the image is uncountably infinite, such as $\Omega_X = \mathbb{R}$, then X is called a *continuous random variable*. For a random variable, X, we refer to its probability distribution as $p(X)$ with the following requirements:

1. $0 \leq p(X = x) \leq 1, \forall x \in \Omega_X$ and
2. $\sum_{x \in \Omega_X} p(X = x) = 1$.

2.1.2 Probability Distributions

Joint Probability Distribution

Given two random variables X and Y, we define the *joint probability distribution* of the two events as follows:

$$p(X, Y) = p(X|Y)p(Y) = p(Y|X)p(X). \tag{2.1}$$

This decomposition is often called the *chain rule*. Given a joint probability distribution, we can account for the influence of all but one variable of choice by computing the *marginal distribution*. This consists of summing over all possible states $y \in \Omega_Y$ for the random variable Y. This operation is called *marginalization* and is formally described as follows:

$$p(X) = \sum_{y \in \Omega_Y} p(X|Y = y)p(Y = y). \tag{2.2}$$

Conditional Probability

When looking at more than one event, X and Y for example, it can be useful to consider the probability of observing state x of X given that we have already observed a certain state of Y. This is called the *conditional probability* of X given Y and is formally denoted as

$$p(X|Y) = \frac{p(X,Y)}{p(Y)}. \tag{2.3}$$

Note that we enforce that for $p(X|Y)$, $p(Y) \neq 0$.

2.2 Comparing Distributions

There are many ways to analyze and compare distributions. For this thesis, focus on two. One that is used when handling categorical features and another when dealing with continuous distributions.

2.2.1 Continuous Distributions

For this investigation we use the *Wasserstein distance*, also known as the *Earth Mover's distance*, when comparing continuous distributions. Given a random variable X and two sampling of that variable X^P and X^Q, the Wasserstein distance between the two samples is precisely the area between the two samples' cumulative density functions $\hat{F}_P(x)$ and $\hat{F}_Q(x)$. The one-dimensional Wasserstein distance (Panaretos and Zemel, 2019) is as follows

$$W_1(X^P, X^Q) = \int_{\mathbb{R}} |\hat{F}_P(x) - \hat{F}_Q(x)| dx. \tag{2.4}$$

Intuitively, the Wasserstein distance describes the total amount of "work" needed to change one distribution into the other. This is aptly visualized if one replaces the continuous distributions with piles of dirt. In this context, the Wasserstein distance would characterize the solution to the mass transport problem of moving shovels of dirt from one pile to the other until the two dirt mounds are identical. This example is the genesis of the name: Earth Mover's distance.

2.2.2 Categorical Distributions

A characteristic of categorical variables is that there is no intrinsic distance between members of the outcome space. What is the distance between the labels dog and cat? For this reason, we can not use the distance metrics which require continuity in the outcome space, such as the Wasserstein distance, when comparing categorical distributions. In order to find an appropriate metric, we look to information theory. Specifically, an symmetrized extension of the *Kullback-Leibler divergence*.

Kullback-Leibler Divergence

As with all our distribution comparison methods, the Kullback-Leibler divergence, or relative entropy, quantifies the difference between two probability distributions. The KL divergence accomplishes this by capturing the amount of excess *Shannon-information* gained by modelling a given probability distribution using a second reference distribution. Shannon-information characterizes the overall uncertainty and

information content of a probability distribution. More precisely, given distributions P and Q of a categorical feature X, KL divergence is defined as

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \ln \left(\frac{P(x)}{Q(x)} \right). \tag{2.5}$$

where $P(x)$ and $Q(x)$ are the probability mass functions of P and Q respectively (MacKay, 2003).

Jensen-Shannon Distance

The main draw back of the KL divergence is that, because it is non-symmetric, it is not a distance measure. Hence, we will use the *Jensen-Shannon distance* when comparing categorical distributions. The Jensen-Shannon distance is the square root of the Jensen-Shannon divergence, which symmetrizes the KL divergence. The JS divergence also benefits from always being bounded, only taking values between 0 and 1, where a divergence of 0 means perfect similarity between distributions and 1 is complete dissimilarity. JS divergence symmetrizes the KL divergence by computing the KL divergence from each distribution with respect to the average of the the two, then takes the average of both divergences. Formally,

$$D_{JS}(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M) \tag{2.6}$$

where $M = \frac{1}{2}(P + Q)$. The JS distance is defined as the square-root of the JS divergence. It benefits from being both symmetric and a proper distance metric. It has shown been shown to be robust against changes in support, meaning it is a good metric to capture drift when novel labels have appeared.

2.3 Data Drift

Given the diversity of terminology within the literature, it is important to be specific about what we define as *data drift*. For the sake of illustration, let us assume that we have perfect knowledge about the classification task of predicting a label y given data X. We train a classification model for this task at time t, using data from the joint distribution $P_t(X, y)$. Now consider we use the model at inference time $t + i$. We say that *data drift* has occurred between t and $t + i$ if:

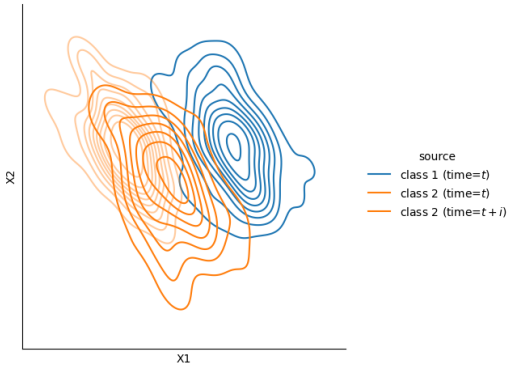
$$P_t(X, y) \neq P_{t+i}(X, y). \tag{2.7}$$

This is broad definition as there are multiple factors which may drive data drift. The chain rule from 2.1 can give us a better insight into the drivers of data drift.

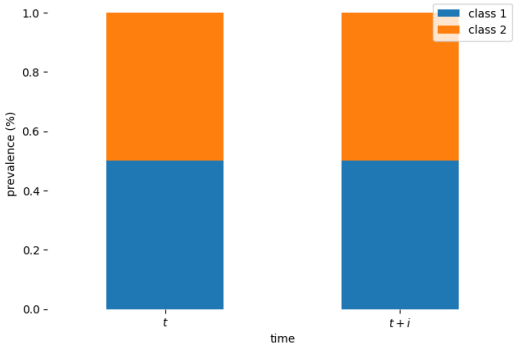
2.3.1 Covariate Drift

The detection of data drift does not always necessitate that our previously learned model will no longer perform as expected. *covariate drift* is a prime example of this and occurs when the data drift is driven by a change in the marginal distribution of X. Formally put, covariate drift occurs when

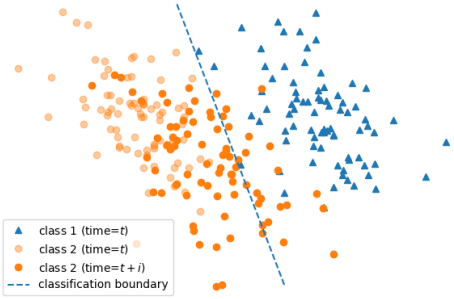
$$P_t(X) \neq P_{t+i}(X). \tag{2.8}$$



(A) KDE plot compares $p_t(X)$ and $p_{t+i}(X)$.



(B) Prevalence plot compares $p_t(y)$ and $p_{t+i}(y)$.



(C) Scatter plot with classification boundary. Compares $p_t(y|X)$ and $p_{t+i}(y|X)$.

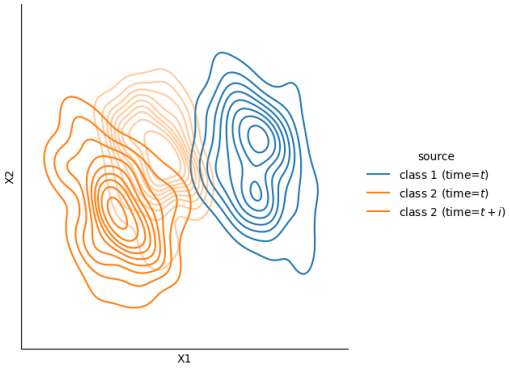
FIGURE 2.1: Example of covariate drift where the classification boundary will no longer capture the relationship between the data and labels. Two time instances are shown, time = t and $t+i$. The initial time = t instance with class 1, class 2, and the learned classification boundary is overlaid with the time = $t+i$ instance.

In Figure 2.1 we can see an example of covariate drift. Initially, a classification boundary between the two classes is learned and does a perfect job at differentiating the two clusters. However, when moving from time t to $t+i$, we see that the distribution of the second class has shifted towards the first class. Notice in Figure 2.1c that the original linear classification boundary no longer perfectly partitions the two classes. The example in Figure 2.1 would show reduced classification performance if the original boundary was continued to be used. This is a example of *model degradation*.

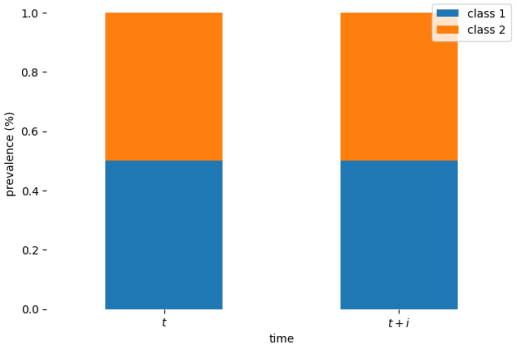
Benign covariate drift, also known as *virtual drift*, refers to covariate drift which has no meaningful effect on a model's inference power. This means that there is a restriction added to the definition of covariate 2.8.

$$P_t(X) \neq P_{t+i}(X) \text{ and } P_t(y|X) = P_{t+i}(y|X). \tag{2.9}$$

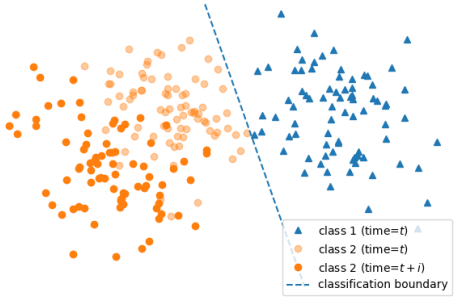
Returning to our example in Figure 2.1, an case of benign covariate drift could be when the two classes drift apart such that the original classification boundary still performs well in its task. This is shown in Figure 2.2. During the occurance of virtual data drift, the remodelling of the problem may not always be necessary. However, just as in any other instance of detected drift, analysis should be conducted to learn more about the factors driving the drift.



(A) KDE plot compares $p_t(X)$ and $p_{t+i}(X)$.



(B) Prevalence plot compares $p_t(y)$ and $p_{t+i}(y)$.



(C) Scatter plot with classification boundary. Compares $p_t(y|X)$ and $p_{t+i}(y|X)$.

FIGURE 2.2: Example of benign covariate drift where the classification boundary continues to capture the relationship between the data and labels.

2.3.2 Label Drift

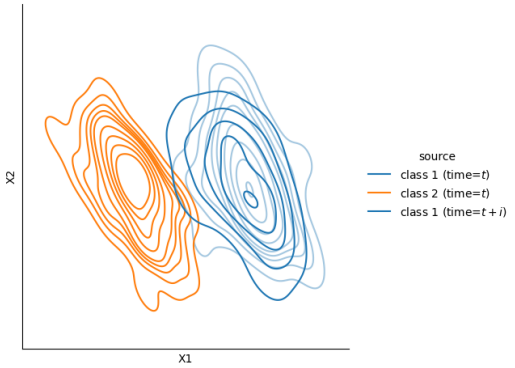
Label drift describes the event when the underlying distribution of targets begins to change overtime, however the distribution of X knowing its target remains stable. Formally, this mean

$$P_t(y) \neq P_{t+i}(y) \text{ and } P_t(X|y) = P_{t+i}(X|y). \tag{2.10}$$

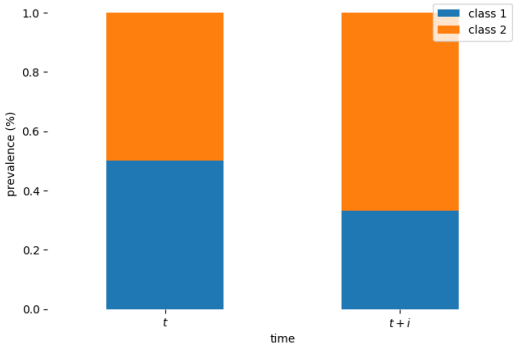
A possible cause of label drift could be a shift in the prevalence of one or more classes. Figure 2.3 shows a reduction of the prevalence of class 1 between time t and $t+i$. Notice that, during label drift, while there is very little change in $p(X)$ and $p(y|X)$, the shift in the label distribution could still lead to a reduction in performance of an AI/ML model. This could happen if the model has learned a bias towards the past label distribution at time t.

2.3.3 Concept Drift

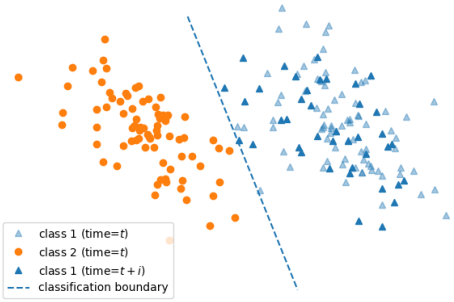
One of the most difficult types of drift to identify, *concept drift* occurs when the distribution of the input data remains stable but the governing relationship between the features X and the targets y is no longer the same. Given that ML and AI models trained on data X at time t will attempt to learn the relationship $P(y|X)$, data drift of this type almost certainly leads to degradation in model performance as the patterns the model has learned to make inferences is no longer descriptive of the relation between the data X and the targets y.



(A) KDE plot compares $p_t(X)$ and $p_{t+i}(X)$.



(B) Prevalence plot compares $p_t(y)$ and $p_{t+i}(y)$.



(C) Scatter plot with classification boundary. Compares $p_t(y|X)$ and $p_{t+i}(y|X)$.

FIGURE 2.3: Example of label drift from time t to $t+i$.

Concept drift is characterized by the following:

$$P_t(y|X) \neq P_{t+i}(y|X) \text{ and } P_t(X) = P_{t+i}(X). \tag{2.11}$$

2.4 Drift Detection

There have been multiple approaches to data drift detection. The predominant techniques can be split into two primary subtypes: *data-distribution based* and *performance-based*, also referred to as *error rate-based*, approaches (Bayram, Ahmed, and Kassler, 2022). Each have their own specific benefits and drawbacks.

2.4.1 Data-Distribution Analysis

Data-distribution based detection approaches put their focus squarely on the data being seen by a given AI/ML model. Through the use of distance measures, they estimate the similarity between data distributions in distinct time-windows. If the distance measure indicates that two distributions are sufficiently dissimilarity from one another, drift is detected. Intuitively, two distributions being significantly dissimilar from one another leads us to believe that the two sample distributions being compared did not come from the same parent distribution. This means that using models trained on old data may perform worse when asked to make inferences on more recent samples.

A benefit of analyzing the data itself is that there are a host of well defined and theoretically sound analysis techniques and metrics for this task. The comparison of sampled distributions is a long standing task within the subject of statistics and

for this boasts a numerous tools for conducting the analysis. For the purpose of this thesis we focus on the *Jensen-Shannon distance* and the *Earth-Mover's distance* (see Section 2.2).

Data-distribution detection methods also work equally well on both labeled and unlabeled datasets, as there is not consideration of the specific model trained on said data. This allows for earlier drift detection in the event that ground-truth collection lags behind data acquisition. This is a common problem when implementing risk predictive models in the clinical setting.

However, these detection techniques are not infallible. There are two major drawbacks that need to be considered when applying a data-distribution based drift detector. The first is such detection techniques perform best when we have perfect knowledge of the underlying distributions that our data is sampled from. As this is impossible in most practical instances, all analyses are conducted between samples and thus are approximations of the true population distribution. Additionally, given that no model performance is being accounted for in these analysis techniques, there is an increased susceptibility to benign covariate drift, as discussed in Section 2.3.1, leading to a heightened risk of false-positive drift detection.

Univariate Analysis Techniques

When conducting a univariate analysis of a dataset, $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ where X_j is a random variable, we consider each feature X_j in isolation. For the purpose of conducting data-distribution drift analysis, this means a direct comparison of distribution of a feature at time t, with the distribution of the same feature at time $t + i$. We are checking to see if there is covariate data drift in each of the dataset's features. This query takes the form

$$p_t(X_j) \stackrel{?}{=} p_{t+i}(X_j) \tag{2.12}$$

and is made for all X_j in our dataset X.

We often find ourselves handling datasets with heterogeneous data types, thus we need methods for comparing both categorical and continuous distributions. In the univariate case, there is many robust statistical tools for comparing distributions and determining their dissimilarity (see Section 2.2).

Multivariate Analysis Techniques

It is often the case that given a dataset, $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, the random variables X_j are not independent of one another. In other words, there are complex relationships which require subsets of $\mathbf{X}$ to be considered in unison. This is were shifting the level of analysis from the individual variables, to the dataset as a whole becomes very useful. For a toy example where a dataset consists of two variables which are highly positively correlated. Now consider that at some point in time, the two features switch from positive to negative correlation but both are still sampled from the same distribution individually. This is a complex covariate drift which only manifests when viewing all the features as they relate to one another. As we can see in Figure 2.4, univariate analysis of the two feature distributions would not be able to detect a substantial difference between the blue and red datasets.

A novel approach to drift detection on the global level (considering inter-variable relationships) is introduced in the drift detection and monitoring python package `NannyML` (*NannyML (release 0.8.6)* 2023). This method consists of splitting comparing two time period's distributions via PCA reconstruction error. A time ordering of the

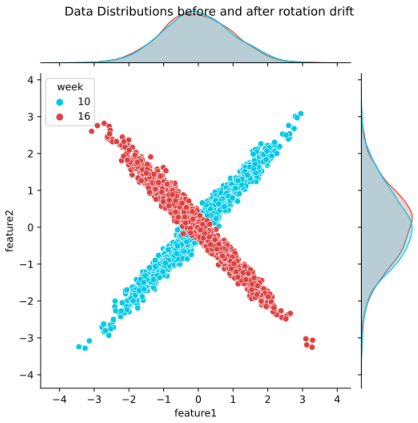


FIGURE 2.4: Example of drift which is only detectable by multivariate analysis. Source: (Nuyttens, 2020).

two distribution is enforced with the former being dubbed the reference set and the latter the analysis set. A PCA is learned using the reference set, then a estimation for the expected reconstruction error is calculated by batching the reference set in time. The reconstruction error is defined as the average Euclidian distance between a data point and the reconstructed position after being projected into and out of the PCA latent space. The PCA projection and reconstruction is applied to each reference batch. This process is never lossless which produces a strictly positive measurement. The mean and variance of these reference reconstruction errors are used to estimate what an expected reconstruction error should be, given the sample is pulled from the distribution that the PCA learned.

The same reconstruction process can be applied to the analysis distribution. If the analysis reconstruction error is significantly different from the expected reconstruction error, then there is evidence of a change in distribution structure between the two time periods. The PCA reconstruction method has been shown to detect the covariate drift in Figure 2.4 and has been used as a baseline comparison for more recent multivarite drift detection methods (Cummings, Snorrason, and Mueller, 2023).

2.4.2 Performance Analysis

The more numerous of the drift detection method subtypes, performance-based drift detectors do exactly as their name suggests: detect model-impacting data drift by monitored performance. These techniques follow the *probably approximately correct learning* model (Mitchell, 1997) and assume that inference error depends on the number of seen examples and the complexity of the hypothesis space. This learning model can be translated into a workable drift detection strategy, as the prior assumptions imply that (given a stationary underlying distribution) the error rate should decrease as the learner is given more examples (Bayram, Ahmed, and Kassler, 2022). Hence, sudden spikes in a learner's error rate may be evidence of the non-stationarity of the underlying distribution, also known as data drift.

The main advantage of performance-based detection strategies over their data distribution-based relatives is that benign data drift will be largely ignored. The data itself is analyzed, but this analysis is done by proxy of the trained model's performance on that data. This forces drift alerts to be triggered by demonstrative degradation in the model's performance, rather than benign changes in the data distribution being sampled. This advantage acts as a double-edged sword. Due to

the model performance being the focus of the detector, the lag between inference and drift detection is based on how quickly ground-truths can be recovered.

Confidence Auditors

In situations where ground-truths are not readily available, the feature space is very large, and rapid drift detection is required, the weaknesses of the prior two detector types are highlighted. The lack of quickly known ground-truths hinders the response time of performance-based detection schemes and the high dimensionality of the data makes data-distribution based detectors computationally cumbersome. This often occurs when a trained AI/ML model is incorporated in a pipeline and inference is made by being fed a constant *data stream*. In these cases a slightly modified version of a performance-based detector scheme may be advantageous. One example of this is the *confidence auditor* detection strategy (Ackerman et al., 2021). Confidence auditors have a defined schema for classification-tasks. For every inference the classifier makes, it will assign a label to the seen data. This assigned label is referred to as the winning label. Similar to the driving idea behind performance-based detectors, confidence auditors operate under the assumption that a significant shift in the distribution of winning label confidences indicate a data drift.

In this described situation, a confidence auditor is able to quickly analyze batches of data. As the detector is only considering the confidence of the model's prediction, not the performance. The auditor has immediate knowledge of the winning label confidence (WLC) at the time of inference. As confidences are bound between 0 and 1, we are able to perform univariate statistical tests on the distribution of batched WLCs. This is much more computationally efficient than performing a multivariate analysis on the whole data distribution and benefits from using a detection metric which is directly tied to the model itself.

Chapter 3

The Data

Our data was sourced from the UK Biobank (UKBB). This data was accessed by application through the EarlyCause project funded by Grant no. 848158 and consisted of 502494 participant evaluations. Data collection began in Stockport in March 2006 (pilot program) and the baseline assessment period continued until October 2010. The assessments recorded a medley of physical, psychological, lifestyle, and exposure data. This information was collected using direct measurement for physical features and a series of questionnaires completed by each participant. These baseline evaluations were conducted in 22 different UKBB assessment centres around the United Kingdom, as see in Figure 3.1.

FIGURE 3.1: Map of all UKBB assessment centres.
Source: UKBiobank External Information

After the baseline assessment period concluded, a repeat assessment period from August of 2012 to June 2013 occurred. These revaluations took place at the Cheadle assessment centre and consisted of a 20345 member subcohort of the original participants. It should be noted not all of the records were re-evaluated during the second assessment period. Hence, only those which were recorded during the second period are considered for this investigation.

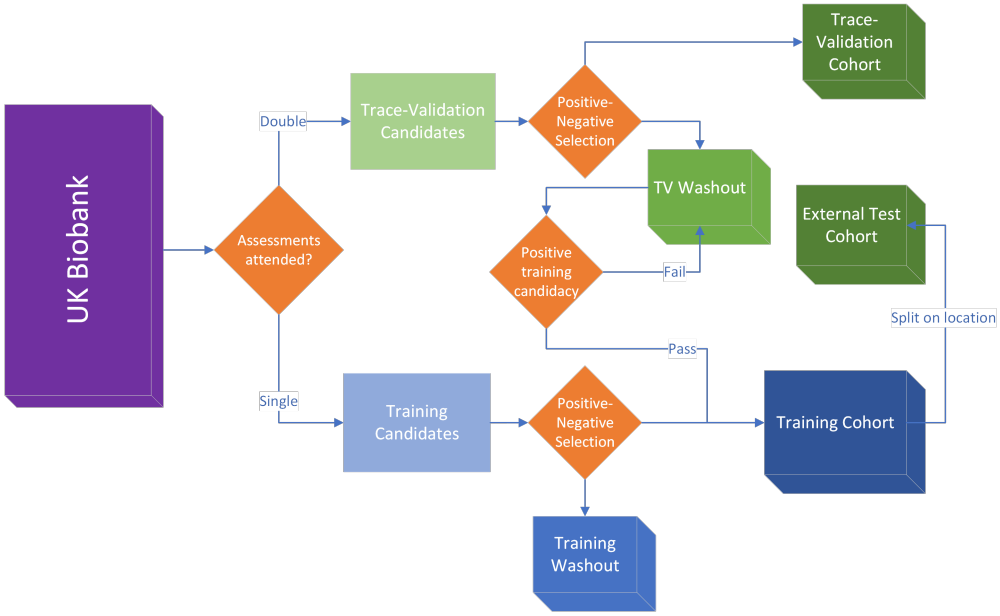


FIGURE 3.2: Selection of training, testing, and traceability cohorts.

3.1 UKBB Exposome Features

Filtering for exposome features, our initial dataset contains 128 features. An exhaustive catalogue of these features is provided in Tables A.1 and A.2 of Appendix A. It is understood that the choice of only exposome features may limit the predictive performance of our models, blood work and other non-exposome features have been shown to be help diabetes risk-predictive power (Mani et al., 2012). However, we make this decision for two reasons. First, acquisition of external exposome data is very cheap, as it does not rely on the use of expensive testing equipment i.e. blood or genetic assays. Additionally, the ability to report this data using self-assessment reduces need for clinical visits to collect the required feature information.

In addition to the diabetes disease group, other groups were explored, namely skin cancer (benign, malignant, and both), breast cancer, lung cancer, prostate cancer, and CVD. The diabetes risk prediction task was ultimately selected. Diabetes mellitus (DM) is the biggest endocrine, and 14th overall, driver for the Global Burden of Disease (Bhutani and Bhutani, 2014). Globally, 45.8%, roughly 74.8 million of all diabetes cases in adults are estimated to be undiagnosed, and nearly 84% of all undiagnosed cases are estimated to be in low- or middle-income countries (Beagley et al., 2014). The vast amount of undiagnosed cases, especially in low-income countries, highlights the need for low-cost (both for the patient and the clinic) risk predictive models which can help direct patients towards proactive care.

3.2 Cohort Selection

When constructing the cohorts, the exposome features were not taken into account. Cohort selection was performed on an individual participant level, by considering the date of the participant's assessment, the location of their attended assessment center, the date of diagnosis and specific ICD-10 codes for participant's diagnosis history. The first step of the process was to split the participants into those who

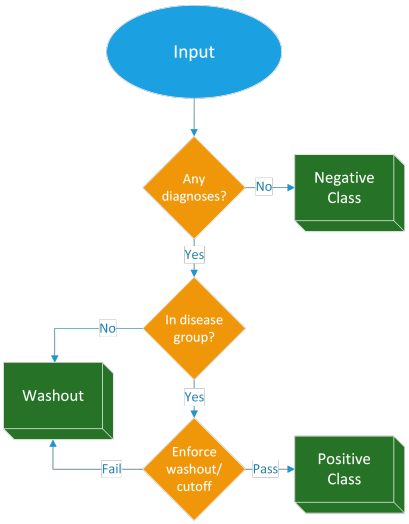


FIGURE 3.3: Positive-Negative selection filter.

attended the follow-up assessment and those who only attended the baseline evaluation stage. Any UKBB participant who attended both assessment periods was considered as a candidate for the time-shifted *traceability-validation cohort*. All others were considered candidates for the *training cohort*. Once the candidates were partitioned, a positive-negative selection filter was applied to each (as seen in Figure 3.3). Three aspects of the participant's candidacy were evaluated by this filter. First, if they had no diagnoses they were selected as members of the healthy (negative) class. The remaining participants were split based on which ICD-10 codes were logged for them (see Table 3.1). If none were a member of the chosen disease group, they were removed from the cohort. Finally a cutoff and washout period was implemented for the remaining positive candidates. Given that our task was risk-prediction, a one year washout period was enforced after the candidates assessment date. This was to ensure that no participant was included who had already been diagnosed before or within one year of their data collection[1]. Diagnosis information was recorded through 2017. Due to this lack of data after 2017, a recency restriction was used in the form of a 5 year diagnosis cutoff enforced from the participant's assessment date. Hence, a candidate assessed on June of 2009 and diagnosed with diabetes in July of 2015, would be removed from the cohort. This enforcement was enacted in order to ensure fair evaluation of performance between the initial assessment period and the follow-up assessment period.

Once the training cohort has been constructed, a portion was split off to form the *external test cohort*. The purpose of this test cohort was for final performance evaluation after model development . The members of this cohort were selected based on the assessment center attended during evaluation. The removed centers were selected so that they were geographically diverse and accounted for roughly 8-10% of participants. This test set evaluated each model's ability to generalize both with respect to unseen data and to populations from unseen locations. The selected assessment centers were Manchester, Oxford, Glasgow, and Cardiff. This selection

[1]To minimize the number of unused participants, those who were washed out of the traceability cohort were passed through the positive-negative selection filter for the training set as well. This means that if a candidate attended the second assessment, but was diagnosed one day prior, they could be added to the positive training class. In this case, no information collected from the second assessment was used.

	ICD-10 Codes	
Diagnosis type	Prefix	Suffix
Insulin-dependent diabetes mellitus	E10	0,1,2,3,4,5,6,7,8,9
Non-insulin-dependent diabetes mellitus	E11	0,1,2,3,4,5,6,7,8,9
Malnutrition-related diabetes mellitus	E12	1,3,5,8,9
Other specified diabetes mellitus	E13	0,1,2,3,4,5,6,7,8,9
Unspecified diabetes mellitus	E14	0,1,2,3,4,5,6,7,8,9

TABLE 3.1: Positive ICD-10 codes for the diabetes disease group.

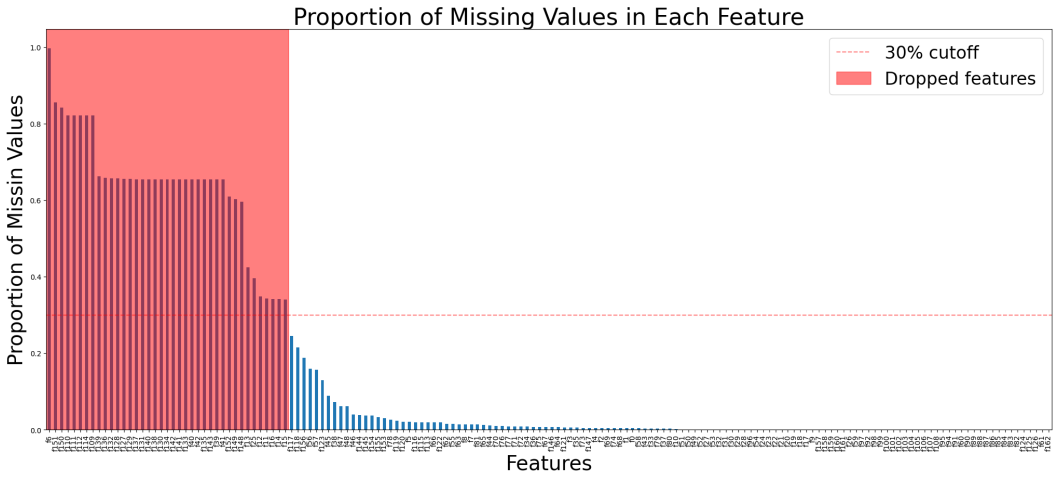


FIGURE 3.4: Total proportion of missing values from each of the 164 transformed features.

ensured we had representation from Wales, Scotland, and both Northern and Southern England.

3.3 Data Cleaning

Prior to cohort selection, Section 3.2, simple data cleaning and feature transformation was required. UKBB uses negative values to encode uninformative or uncertain labels. The values -3 or -818 are often used to encode the "Prefer not to answer" response -1 often encodes a response of "Unsure" or "Do not know", where as -2 might indicate that the question does not apply to the participant. In these cases we replaced such responses with `NaN` based on the data encoding scheme used for that specific feature. This was done so that data imputation could be performed to estimate these values.

After the data was cleaned, one-hot encoding was applied to many of the categorical features. This increased the size of our feature space from 128 to 163. Once the one-hot encoding was complete, any feature with greater than 30% of its values being `NaN` were dropped as they would require too much imputation to be considered reliable features. Figure 3.4, shows all the proportional missing values for each feature in the 163 dimensional feature space. In total 39 features were considered to be saturated with missing values and were dropped. This reduced our feature space

Future Diagnosis	Training Cohort	Test Cohort	Traceability Cohort
No	77305	10982	3714
Yes	8350	1033	293
Totals	85655	12015	4007

TABLE 3.2: Final class membership totals for each cohort

to its final size of 124 features. The next step was to apply a second 30% missing data filter, this time to the each of the remaining cohort members. This resulted in the removal of 202 members of the training cohort, 10 members of the external test cohort, and 2 participants of the traceability cohort. This marked the end of the data cleaning routines. The final cohort sizes can be found in Table 3.2.

Chapter 4

Model Development

4.1 Task Definition

Before proceeding, it is prudent to clearly put forth the task our models will be ask to perform. Using the only the features in Chapter 3, we ask our models to differentiate between healthy individuals and those whom will be diagnosed with diabetes within the a five year period.

4.2 Model Architectures

We briefly outline the model architectures evaluated for the risk predictive classification problem.

4.2.1 Logistic Regression

Logistic regression is a popular family of discriminative machine learning model. The model seeks to distinguish between possible classes by assuming a linear relationship between the input features and the log-odds of the outcome. By using a logit (sigmoid) function, logistic regression maps the linear prediction to a probability score between 0 and 1, representing the likelihood of the positive class. Consider a finite set of input features, X, and a binary target label y. Let π be the log-odds of y being the positive label. Then the logistic function applied to π is

$$\text{logit}(\pi) = \frac{1}{1 + e^{-\pi}} \tag{4.1}$$

where

$$\pi = \beta_0 + \sum_{x_i \in X} \beta_i x_i \tag{4.2}$$

Training a logistic regression model involves estimating the optimal coefficient values of the β_is. This is typically done using gradient descent.

Logistic regression offers interpretability, as the coefficients associated with each feature indicate their impact on the outcome. It has been used for diabetes prediction (Joshi and Dhakal, 2021) and has been shown to perform on par or better with most machine learning models for clinical predictions (Christodoulou et al., 2019). However, it should be noted that logistic regression can suffer from overfitting, particularly when asked to make predictions using a high number of features (IBM, 2021).

4.2.2 Support Vector Machines

Support Vector Machines (SVMs) are a family of machine learning models used for binary classification. SVMs aim to find the optimal decision boundary that maximally separates the two classes in the feature space. The decision boundary is determined by a subset of the training samples, called support vectors, which lie closest to the boundary. SVMs have the ability to handle high-dimensional data and are robust to overfitting. They can also find non-linear decision boundaries through the application of the kernel trick. This consists of mapping features into a higher-dimensional space through a kernel function. In this higher-dimensional space, the SVM linearly separates the two classes and then maps the decision boundary back to the original feature space. Hence, SVMs are able to describe non-linear relationships. However, the choice of kernel function dictates the type of non-linear boundary the SVM can define. Support vector machines have a long history of use for the task of diabetes diagnosis and prediction (Kavakiotis et al., 2017).

4.2.3 Random Forest Ensembles

Random Forest is a popular ensemble method for binary classification tasks. It democratizes the prediction process by considering the predictions of multiple decision trees. Each decision tree in the ensemble is trained on a random subset of the training data and a random subset of features. Using the Law of Large Numbers, the algorithm has been shown to converge without overfitting (Breiman, 2001). When performing inference, each tree independently classifies the input and the final prediction is made using some decision strategy. For classification tasks this strategy is often majority vote. Random Forests can handle high-dimensional data and are capable of capturing non-linear relationships between features and the target variable. They are also less sensitive to outliers and noise compared to individual decision trees. However, all the decision trees in an ensemble are trained independently of one another and greedily. There is also the physical storage concern if the pattern that the ensemble is trying to learn is very complex. This is due to the increased depth of each tree required to form complicated rules (Ren et al., 2015)

Balanced Random Forest

An issue can arise when training random forests on imbalanced data. Given that the training data for each independent decision tree is constructed using bootstrapping, if our data is highly imbalanced then some trees run the risk of being trained on samples with little to none minority class representation. This results in that part of the ensemble not being useful in classifying the minority class. The ramifications of this risk is heightened when the classification task is in the clinical setting (disease prediction or diagnosis) as the label which would require intervention on the clinic's part is usually the minority class (positive member of the disease group). To combat this, the Balanced Random Forest (BRF) algorithm is proposed in Chen, Liaw, Breiman, et al., 2004. This alteration to the random forest ensemble enforces that the sampled training data must be balanced on the minority class. Succinctly put, the algorithm consists of three steps:

1. For each iteration in the random forest, draw a bootstrap sample from the minority class. Randomly draw the same number of cases, with replacement, from the majority class.

2. Induce a classification tree from the data to maximum size, without pruning.
3. Repeat the two steps above for the number of times desired. Aggregate the predictions of the ensemble and make the final prediction.

Both random forest and balanced random forest have been used for early diabetes prediction (Shrivastava et al., 2022).

4.2.4 Gradient Boosting

Gradient boosting is a machine learning technique that combines multiple weak predictive models, to form a stronger and more accurate one. First proposed by Friedman, 2001, it differs from other ensemble techniques by - rather than learning each classifier independently - iteratively fitting each member of the ensemble to the residuals of the prior member. The predictions of all the models are aggregated together to produce the final prediction. This iterative process continues until a stopping criterion, such as a maximum number of models or a minimum improvement threshold, is met.

XGBoost

The eXtreme Gradient Boosting algorithm (commonly known as XGB or XGBoost) is a widely used gradient-boosting algorithm known for its exceptional performance in a variety of tasks, including classification, regression, and ranking (Chen and Guestrin, 2016). It is based on the gradient boosting framework and employs an ensemble of weak decision trees to make accurate predictions. The reason for its popularity within the data science community is due to the combination of high-end performance with efficient training speeds. Due to the large complexity of XGB ensembles, regularization is usually required to dampen the risk of overfitting.

4.3 The Learning Scheme

4.3.1 Model Pipelines

The training pipeline comprised several essential steps to prepossess and optimize the input data before training the final model. It began with missing value imputation. As all our data now has a numeric encoding, we replaced missing values with the median of that respective feature. This was done to ensure there was no data leakage due to imputed data. When training, each model learns its own imputation scheme only from the seen data. Given the high imbalance between classes, we applied random under sampling of the majority class along with a synthetic data augmentation of the minority class using `SMOTE` (Chawla et al., 2002). The latter technique helps to address class imbalance by generating synthetic examples based on the characteristics of the minority class. Lastly, the pipeline selected the 15 best features by ranking them according to their ANOVA f-statistic. This was implemented using the `SelectKBest` function from `scikit-learn` package (Pedregosa et al., 2011). This input feature restriction was recommended in conversations with clinicians. All discussed techniques are recommended when dealing with imbalanced datasets (Kotsiantis, Kanellopoulos, Pintelas, et al., 2006). Each step in the pipeline is given trainable parameters to create a comprehensive training pipeline that improves the quality of the data, handles class imbalance, and selects the most relevant features, ultimately leading to an optimized model.

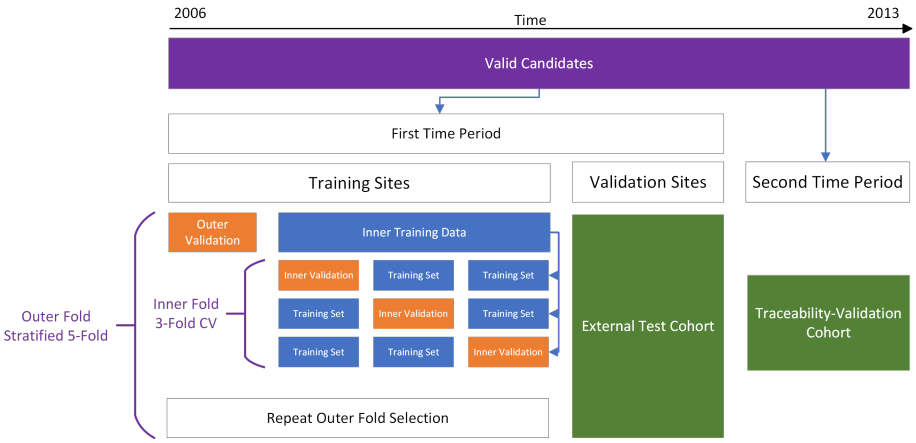


FIGURE 4.1: Nested cross-validation scheme used for hyperparameter tuning.

4.3.2 Nested Cross-Validation

After pipeline construction, we approached the training process using a nested cross-validation scheme. This scheme followed a 3-5 inner-outer fold split. The training data was divided into five stratified partitions. Folds of this stratified partition were constructed such that one of the partitions was removed to act as the *outer validation set*. The remaining four partitions were combined to for the *inner training fold*, which was used to optimize the hyperparameter configuration. To tune the parameters, each inner training fold was then partitioned using a 3-fold cross-validation scheme. We evaluated all configurations from a 50 member sample of the defined hyperparameter space. This random selection and evaluation was repeated once for each inner fold. The parameter configuration which performed best over all folds in the 3-fold cross-validation was then retrained on the entire inner training fold and passed back to the outer fold. Once passed to the outer fold, a final decision threshold was tuned using the left out portion of the outer stratified partition. The partitioning of all data is depicted in Figure 4.1.

Tuning Metrics

For the inner cross-validation, average precision (AP) was used for the tuning metric. AP is a metric which summarizes the precision-recall curve. It differs from calculating the area under the precision-recall curve, by calculating the weighted mean of precisions achieved at each decision threshold, with the increase in recall from the previous threshold used as the weight. That is,

$$AP = \sum_n (\text{recall}_n - \text{recall}_{(n-1)})\, \text{precision}_n \tag{4.3}$$

where precision_i and recall_i are the precision and recall scores for the i^{th} decision threshold. When dealing with imbalanced datasets the precision-recall curve is often more descriptive than evaluating the receiver operator curve (Saito and Rehmsmeier, 2015). Unlike other common tuning metrics, like f1-score, AP benefits from being decision threshold agnostic. This means that the metric is a consideration of all possible decision thresholds. Metrics which lack this quality require a decision threshold to be defined before they can computed. As we tune the decision threshold in the outer fold this threshold agnosticism is a very useful property.

Once a configuration of hyperparameters has been selected, the decision threshold was tuned using the left out partition in the outer cross-validation. This tuning was done with regards to the f1-score, which is the harmonic mean between the precision and recall scores.

$$\text{f1}_{score} = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{precision}} \tag{4.4}$$

This allowed us to consider both the *positive predictive value* and the *true positive rate* when selecting a final decision threshold.

Chapter 5

Performance Results

5.1 Model Performances

In order to estimate model performance, we evaluated each outer fold model. By taking the mean and variance of all outer fold performances for a given architecture, we were more confidently able to estimate true performance. In total, we evaluated on four performance metrics. As stated in Section 4.3.2, *average precision* and *f1-score* are evaluated for each model. In addition to the prior two metrics, *precision* and *recall* were also evaluated. Note that only average precision is a threshold agnostic metric, meaning f1, precision, and recall were all calculated using the optimized decision threshold selected during the model development process.

Figure 5.1 compares the models' performance on the training and external test cohorts. Note that the BRF classifiers seem to be overfitting on the training cohort, resulting in a steep drop in overall performance when making inference on the external test cohort. SVM performed poorly on both cohorts. XGBoost and Logistic Regression were the best overall performers on these two cohorts. The metrics of each have very little change between the training and external test sets which led us to believe these architectures were best able to generalize to unseen data and data collected at novel locations (universality). Figure 5.2 makes the same comparison, now between the external test cohort and the traceability cohort. For exact performance estimations, see Table 5.1. We found worse performance across every architecture and for every metric. The smallest performance drop for the XGBoost and logistic regression architectures was 10% (see Table 5.2). This was a strong indication that model degradation occurred sometime in between the baseline assessment period and the follow-up assessment period.

We sought to understand the driving factors for this stark degradation and decided to perform a distribution-based data drift analysis on the best performing feature subsets. Having both logistic regression and XGBoost performing similarly well, we choose to focus on the best performing models for each architecture.

5.2 Best Model Selection

As we allowed each model to learn its own input feature space, we needed to choose a representative model for the two architectures. Using the training cohort performance, we selected the best model by AP score. The Receiver Operator Curve (ROC) has commonly been used in the development of clinical risk assessment models for diabetes (Buijsse et al., 2011). The ROC plots the recall against the false-positive rate over all possible decision thresholds and is most commonly summarized by the area underneath this curve. This summary of the ROC reports the probability that the predicted risk for a positive subject is higher than that for a negative participant and

FIGURE 5.1: Estimations of architecture performances on Training and Test cohorts. Evaluation metrics include: Average Precision, F1-score, Precision, and Recall.

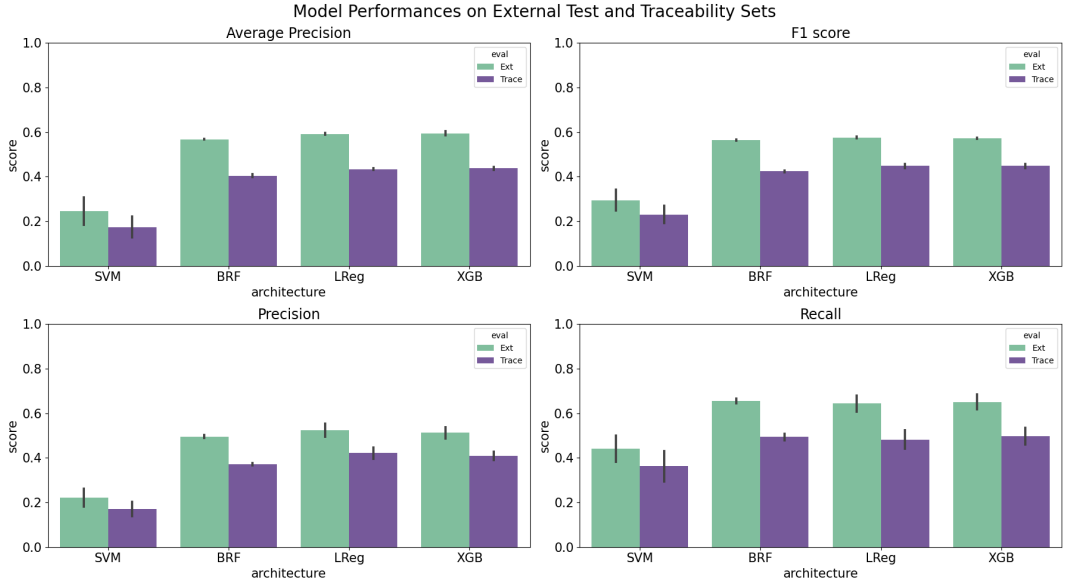


FIGURE 5.2: Estimations of architecture performances on External Test and Traceability-Validation cohorts. Evaluation metrics include: Average Precision, F1-score, Precision, and Recall.

Architecture	Avg. Precision	F1	Precision	Recall
SVM				
Train	0.248 ± 0.044	0.301 ± 0.039	0.238 ± 0.039	0.411 ± 0.031
Test	0.246 ± 0.055	0.295 ± 0.042	0.222 ± 0.036	0.441 ± 0.052
Trace	0.174 ± 0.042	0.231 ± 0.035	0.170 ± 0.028	0.362 ± 0.061
BRF				
Train	$\mathbf{0.832 \pm 0.003}$	$\mathbf{0.723 \pm 0.005}$	$\mathbf{0.630 \pm 0.011}$	$\mathbf{0.849 \pm 0.007}$
Test	0.568 ± 0.002	0.564 ± 0.003	0.495 ± 0.006	0.655 ± 0.010
Trace	0.404 ± 0.006	0.424 ± 0.004	0.372 ± 0.005	0.494 ± 0.013
Log Reg				
Train	0.566 ± 0.004	0.551 ± 0.004	0.523 ± 0.023	0.585 ± 0.033
Test	0.592 ± 0.004	$\mathbf{0.576 \pm 0.003}$	$\mathbf{0.524 \pm 0.026}$	0.643 ± 0.033
Trace	0.434 ± 0.003	$\mathbf{0.448 \pm 0.008}$	$\mathbf{0.421 \pm 0.024}$	0.483 ± 0.037
XGBoost				
Train	0.580 ± 0.005	0.560 ± 0.003	0.521 ± 0.016	0.607 ± 0.027
Test	$\mathbf{0.594 \pm 0.008}$	0.573 ± 0.003	0.513 ± 0.023	$\mathbf{0.651 \pm 0.031}$
Trace	$\mathbf{0.438 \pm 0.006}$	$\mathbf{0.448 \pm 0.009}$	0.409 ± 0.016	$\mathbf{0.498 \pm 0.034}$

TABLE 5.1: Model performance by architecture, cohort, and evaluation metric. Bold values mark the best performance for that fixing metric and cohort.

	Performance Drop (%)			
Architecture	Avg. Precision	F1	Precision	Recall
Log Reg	15.78 ± 0.23	12.82 ± 0.85	10.29 ± 0.80	16.08 ± 0.82
XGBoost	15.65 ± 0.61	12.45 ± 1.08	10.33 ± 1.24	15.37 ± 0.63

TABLE 5.2: Percentage drop in performance from external test to traceability cohort for logistic regression and XGBoost architectures.

Architecture	Avg. Precision	F1	Precision	Recall
Best Log Reg				
Train	0.573	0.558	0.507	0.620
Test	0.599	0.575	0.497	0.682
Trace	0.440	0.462	0.407	0.536
Best XGBoost				
Train	0.587	0.563	0.509	0.628
Test	0.605	0.572	0.495	0.678
Trace	0.445	0.448	0.392	0.522

TABLE 5.3: Performance metrics of best logistic regression and XGBoost models.

is a good proxy for the overall goodness of the model. However, it has been proposed that AP is a more appropriate metric when evaluating risk predictive models for low prevalence disease groups (Su, Yuan, and Zhu, 2013).

Using the training cohort AP score as our selection metric, we found that both the highest performing logistic regression and XGBoost models had selected the same feature subspace. Moving forward, we only analyzed possible data drift within this feature space. The performances of the two selected models on all three cohorts are reported in Table 5.3.

Chapter 6

Data-Distribution Drift Analysis

We performed two levels of data-distribution drift analysis. The first was a multivariate analysis to check for covariate data drift on the global scale. We followed this macro scale analysis with univariate analyses to detect covariate drift within the individual features. This analysis was done using implementations from the `NannyML` python library. We only investigated the subset of our feature space determined by the best performing Logistic Regression and XGBoost models. That subset is presented in Table 6.1.

6.1 Multivariate Analysis

We implemented the multivariate PCA reconstruction method outlined in 2.4.1. As both the training and external test cohorts come from the same time period, their participants were combined to form the reference set. The analysis set consisted of only the traceability cohort.

We first learned a PCA embedding which captured a minimum of 70% of the variance within our reference set. We then needed to learn the expected reconstruction error and tolerance for deviation. This was done by ordering all participants in the reference set by date of first assessment, then partitioning them into batches of 500 members. The PCA reconstruction method was applied to each of the batches and their respective reconstruction errors were calculated. Our mean expected reconstruction error was around 1.835 with an upper and lower tolerance of 1.993 and 1.676 respectively. The upper (or lower) tolerance were defined as the mean expected error plus (or minus) three standard deviations. None of the reference batches fell outside these tolerances. Any batch whose reconstruction error fell outside of these bounds would be considered evidence of covariate data drift's presence.

We then applied the same batching process to the analysis set. The mean reconstruction error of the analysis batches was 1.713 with a standard deviation of 0.057. Two of the analysis batches fell outside of the reconstruction error tolerance. The analysis batch from October 28th to December 3rd 2012 had a reconstruction error of 1.645 and the analysis batch from January 16th to Febuary 22nd 2013 had a reconstruction error of 1.650. Figure 6.1 shows the reconstruction errors of both the reference and analysis batches along with the drift alert thresholds and analysis batches which exceeded those tolerances.

6.2 Univariate Analysis

Following the multivariate PCA reconstruction analysis, we conducted data-distribution analyses for each feature individually. We used both the training and external test

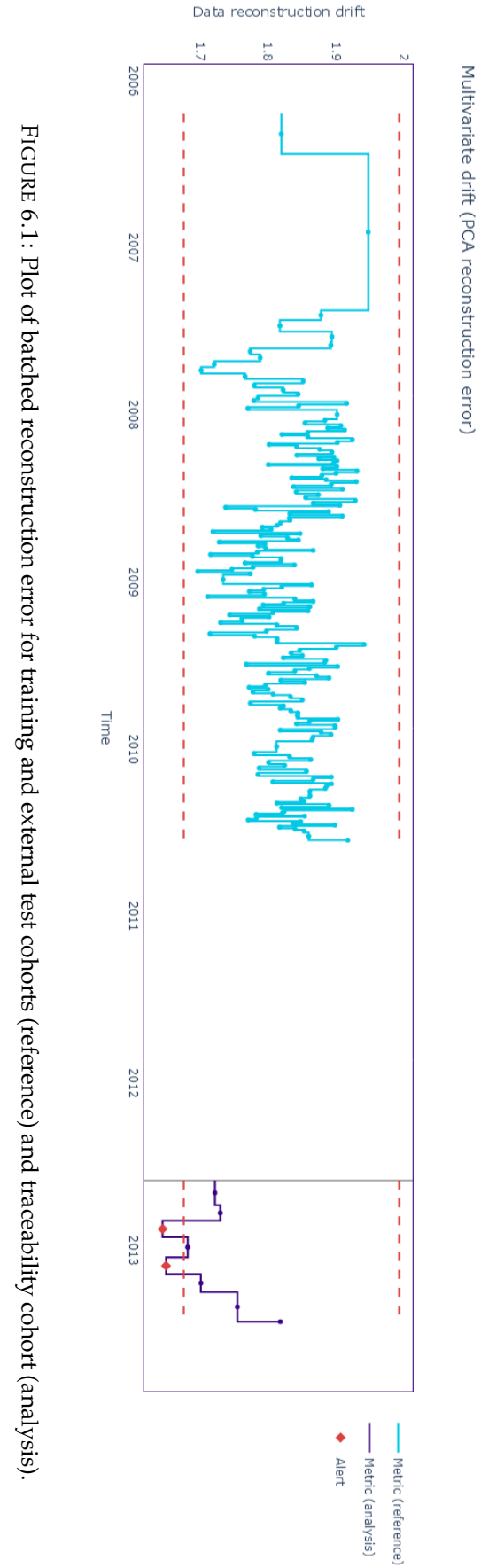


FIGURE 6.1: Plot of batched reconstruction error for training and external test cohorts (reference) and traceability cohort (analysis).

Feature Types	
Categorical	Continuous
Employment Status	Waist Circumference
Retirement Status	Hip Circumference
Ability to Work	BMI
Lack of Education	Weight
Frequency of Day Naps	Whole Body Water Mass
Diet Change in Prior 5 Years	Basal Metabolic Rate
Diet Change due to Illness	Birth Year
Frequency of Tiredness	

TABLE 6.1: Feature subspace selected by best performing models.

cohorts as the reference set and the traceability cohort as the analysis set. For continuous features, the Wasserstein distance was used (see Section 2.2.1). The same partitioning scheme from the multivariate case was followed for the time-ordered reference and analysis sets. The appropriate Wasserstein distance was calculated between each of the reference batches and the whole reference set. The mean and standard deviation of the distances were used to set the distance tolerance for the analysis set. As all distance metrics are non-negative in value, only the upper threshold was required. The upper threshold was defined as three standard deviations above the mean distance. For categorical features, the Jensen-Shannon distance was used (see Section 2.2.2) as the distance metric. As the JS distance is bound between 0 and 1, a fixed value of 0.1 was used as the upper threshold. The same partitioning was done on the analysis set and batches whose JS distance exceeded 0.1 were considered to be drifting.

All plots generated for this analysis are included in Appendix B. Figures B.1 and B.2 shows the analysis batch alert and distribution plots for all categorical features. The only categorical features to register drift are Lack of Education and Frequency of Tiredness. Figures B.3 and B.4 shows the analysis batch alert and distribution plots for all continuous features. Of the continuous features, BMI, Basal Metabolic Rate, and Birth Year have batch Wassertein distances which exceed their features upper tolerance. However, the batch which triggered a drift alert for Basal Metabolic Rate contained few participants and as such we ignored this alert moving forward.

6.3 Confidence Analysis

We also conducted an analysis of both the best Logistic Regression and XGBoost model's predictions and confidences for each of the partitioned batches. We use the proxy feature of positive class probability for the prediction confidence. Figure 6.2 shows the combined analysis batch alert and distribution plots for the Logistic Regression model. As we are dealing with a binary classification task, the predictions are considered a categorical feature and the positive class probability is continuous. There were not any batches which exceeded the upper threshold for either feature.

Figure 6.3 shows the combined plots for the XGBoost model. Similar to the Logistic Regression model, the distribution of predictions did not significantly differ from

the reference set. However, there was a batch which exceeded the threshold for the confidence of predictions. In addition, two other batches' Wasserstein distance were very close the feature's upper tolerance.

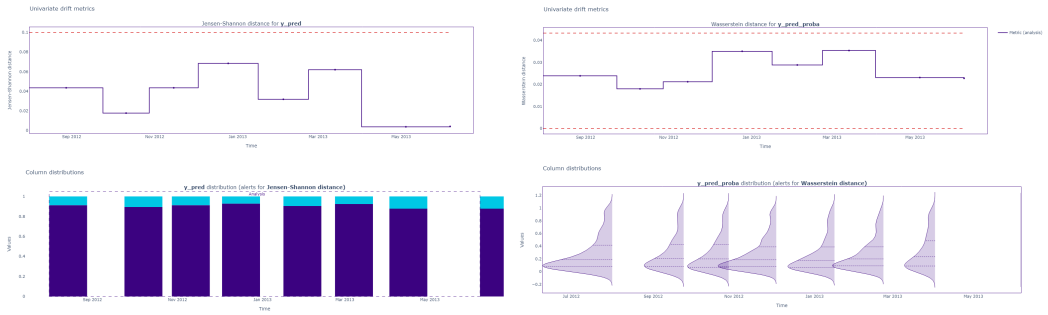


FIGURE 6.2: Combined analysis batch drift alert and distribution plots for the best Logistic Regression model.

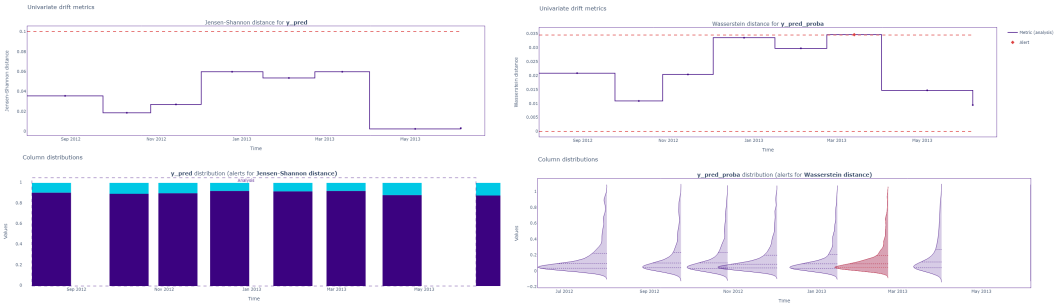


FIGURE 6.3: Combined analysis batch drift alert and distribution plots for the best XGBoost model.

6.4 Prevalence Rate Analysis

In order to check for potential label data drift, the true diagnosed prevalence rate was recorded for each of the batches used in the multivariate PCA reconstruction. The same reference and analysis split was made and an upper and lower tolerance for the diagnosed prevalence rate was set as the mean reference prevalence rate plus and minus three standard deviations. Figure 6.4 shows the batch-wise prevalence rate for both reference and analysis set. While the mean diagnosed prevalence rate for the analysis set was lower than the reference set's, it was not outside of our set tolerances.

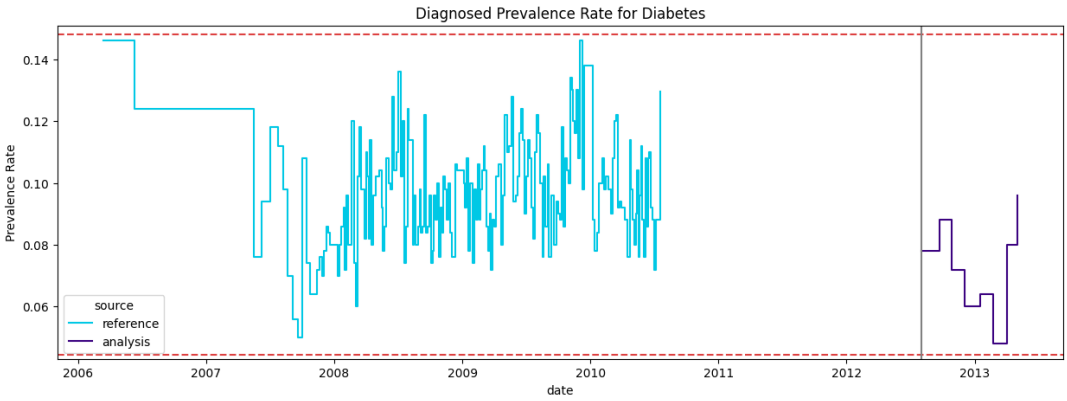


FIGURE 6.4: Batch-wise diagnosed prevalence rate for the reference and analysis sets. Upper and lower tolerance for deviation from the mean reference prevalence rate are marked in red.

Chapter 7

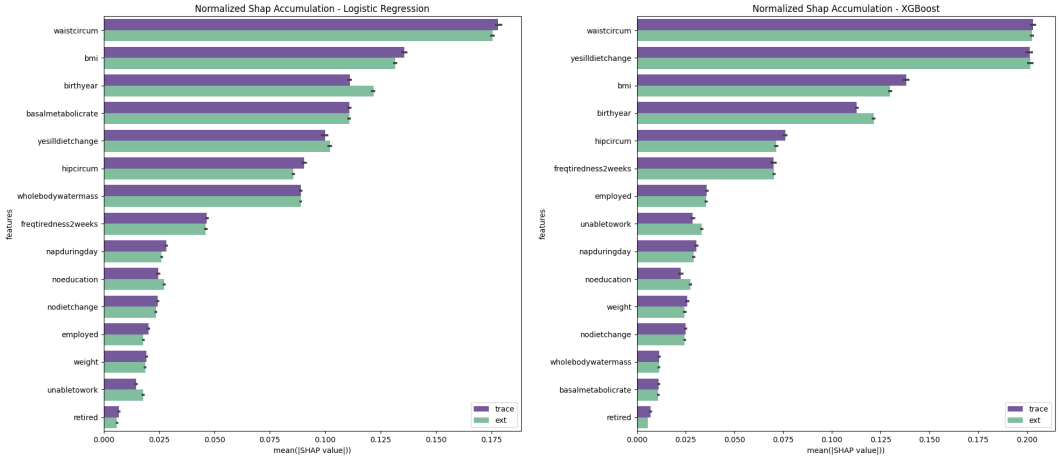
SHAP Importance Drift Analysis

In Chapter 6, the analysis was restricted to the data-distributions and the model predictions themselves. The former ignores all model input and the latter does not take into account the contributions of the input features. In order to build a better understanding of how the how the best Logistic Regression and XGBoost models were using the input features we turn to analysing the absolute normalized Shapley ($|\,\text{nSHAP}\,|$) contributions of each feature. Error estimations for the $|\,\text{nSHAP}\,|$ contributions were calculated via a 5-fold split of each cohort's participants. The $|\,\text{nSHAP}\,|$ contributions were calculated for each feature on each fold and the mean and standard deviations were reported.

For each model, two forms of comparisons were conducted. First, the $|\,\text{nSHAP}\,|$ contributions for each feature were calculated for the external test cohort and the traceablility cohort. A feature-wise comparison was done to identify the features which had the most significant absolute contribution shift. The second comparison, inspired by (Duckworth et al., 2021), focused on the temporal change in $|\,\text{nSHAP}\,|$ contribution. To make this comparison, we ordered all participants by date of assessment and then partitioned them using the same batching scheme from Chapter 6. The reference set was composed of all batches coming from the training or external validation cohorts and the analysis set consisted of batches pulled from the traceability cohort. The $|\,\text{nSHAP}\,|$ contribution of each feature was computed for every batch and a 5-fold split was used to estimate the contribution error. Batches with fewer than 30 participants were dropped to avoid inflated variance. The mean $|\,\text{nSHAP}\,|$ contribution and standard deviation for each feature was reported for all remaining batches. Upper and lower feature contribution tolerances were set as the mean of all contributions for the reference set batches plus and minus three standard deviations.

Logistic Regression		**XGBoost**	
Top 5 Features	$\Delta\|\,\text{nSHAP}\,\|$	**Top 5 Features**	$\Delta\|\,\text{nSHAP}\,\|$
Birth Year	0.0107	Birth Year	0.0087
Hip Circumference	0.0047	BMI	0.0082
BMI	0.0042	Lack of Education	0.0050
Ability to Work	0.0032	Hip Circumference	0.0047
Lack of Education	0.0027	Ability to Work	0.0045

TABLE 7.1: Top 5 features by change in mean $|\,\text{nSHAP}\,|$ contribution for Logistic Regression and XGBoost.



(A) |nSHAP| contributions (Logistic Regression). (B) |nSHAP| contributions (XGBoost).

FIGURE 7.1: Comparison plots of |nSHAP| contributions for Logistic Regression and XGBoost models.

7.1 Logistic Regression Model

The first method of comparison for the Logistic Regression model is presented in Figure 7.1a. The absolute difference between the external test and tracebaility cohort |nSHAP| contribution and the top five features with respect to absolute change in contribution were recorded. Table 7.1 given the top five features and their respective absolute difference in contributions.

The second, time-ordered, comparison was used to detect gradual shifts in the feature contribution over time. All drift alerts from the reference set are ignored as we are interested in deviations in the analysis set. The only feature to trigger a |nSHAP| drift alert is Birth Year, while Weight, Employment Status, and Frequency of Day Naps come close to triggering alerts. The names of those features who registered drift in Section 2.3.1 are italicized and the top five features from the prior comparison are in bold. Interestingly, only one feature which registered possible covariate drift did not experience a difference in mean |nSHAP| contribution between the reference and analysis sets. The results of this comparison are shown in Figure 7.2.

7.2 XGBoost Model

The first SHAP comparison for the XGBoost model is shown in Figure 7.1b. The top five features in terms of difference in |nSHAP| contribution between the external test and traceability cohorts recorded and the change in mean |nSHAP| contribution for these features can be seen in Table 7.1. We found that the top five feature were the same as those found for the Logistic Regression model, albeit the ordering of the features and magnitudes of their respective changes differed.

Figure 7.3 shows the temporal comparison of the mean monthly contributions for each feature. Interestingly, none of the batched analysis feature contributions trigger |nSHAP| drift alert. However, Birth Year and Whole Body Water Mass have confidence intervals which exceed the contribution deviation tolerances. Plots with bold titles indicate the feature was one of the top five features for change in mean

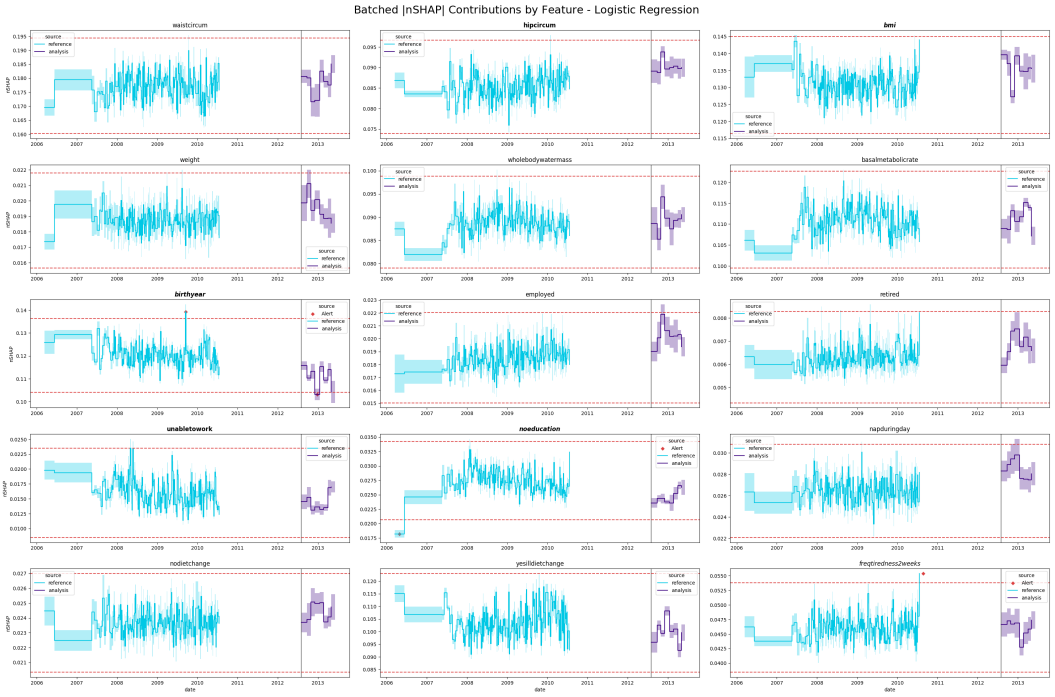


FIGURE 7.2: Temporal comparison of monthly |nSHAP| contributions for Logistic Regression.

|nSHAP| contribution. Plots whose titles are italicized are features whom registered drift alerts during Section 6.2.

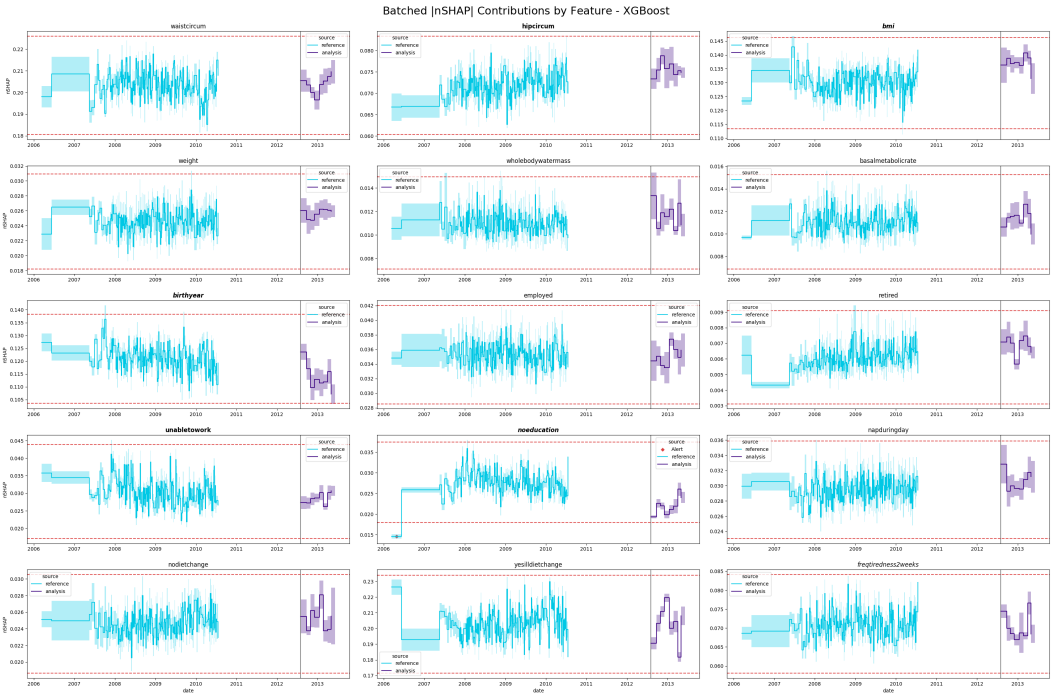


FIGURE 7.3: Temporal comparison of monthly |nSHAP| contributions for XGBoost.

Chapter 8

Discussion

All the trained models experience a stark drop in performance when making inference on a cohort which is temporally distant from the period that training data was collected. The assessments for members of the traceability cohort were conducted at a location which was not seen in the training data. However, we believe we may rule out location bias as a driver of the degradation considering the external test cohort, whose assessment centres were also removed from the training data, showed much higher performance. We see this as strong evidence that the cause may be connected to one or multiple types of data drift.

In Chapter 6 we investigated this hypothesis. We found that portions of the best performing feature subset had extreme PCA reconstruction errors. Surprisingly, the reconstruction error was significantly lower than that of our training cohort data. This implies that the PCA which was used to estimate the reconstruction error was a better representative of the traceability cohort (analysis set) than the training and external test cohorts (reference set) on which it was learned. It was expected that significant deviations from the expected reconstruction error would have higher errors than the reference set. The reason for the analysis set being less effected by passing through the PCA latent space is unknown and worth further investigation. Four features were identified as possibly experiencing covariate data drift between 2010 and 2012, namely Lack of Education, Frequency of Tiredness, Birth Year, and BMI. We are very interested to see if a gradual shift can be seen in these feature's distributions over time. However, due to a lack of assessment data between those dates, this may be beyond our abilities.

A performance-aware drift detection was run via a proxy for model prediction confidence. This revealed that only the XGBoost model demonstrated a significant deviation in its positive class probability distribution. It is of interest, to reformulate this test and directly use model prediction confidence. Section 6.4, found no potential label drift alerts. Considering the heavy imbalance between the positive and negative classes, small changes in the batch prevalence may be impacting model performance even if the changes are within our reference set deviation tolerance.

In Chapter 7, we used Shapley values to better understand how the models were using each of the input features. These Shapley contributions were compared between the external test and traceability cohorts. In this comparison we found multiple differences in the feature contributions and both models exhibited similar changes in said contributions. They were both found to have the same five feature experience the largest contribution shift. Both models showed the relative contribution of BMI and Hip Circumference drop, while the relative contributions of Birth Year, Lack of Education, and Ability to Work increased. Both models registered the same total change in absolute feature contribution. However, the Logistic Regression model exhibited higher contribution shifts for the most shifting features than

the XGBoost model. Interestingly, three features are shared between the most significant changes in feature contribution and the set of features who indicated possible covariate data drift. This could be an indication of data drift driving the model degradation.

A time-evolution of Shapley values was also constructed. Due to the lack of information between the years of 2010 and 2012, it is difficult to identify specific trends or seasonality within the batched feature contributions across time. However, the observation that Logistic Regression experienced a larger change in its most shifting feature is supported as it is the only model which registered $|\text{nSHAP}|$ drift alerts within the analysis set. Another observation is that the XGBoost model seemed to consolidate more absolute feature contribution in fewer of the input features than Logistic Regression. This might explain XGBoost being the only model which triggered an analysis set drift alert for the distribution of positive class probabilities in Section 6.3. To the best of our knowledge, while SHAP analysis have been proposed to detect drifts for medical monitoring models, this is the first study to propose it for a disease risk prediction task.

Lastly, there are multiple factors that may have led to the traceability cohort being biased. This is due to the reassessments being voluntary and performed in one location. This may produce a representation bias towards individuals with more flexibility with their time and ability to travel. This bias may be driving the data drift found during this investigation. As the reassessment cohort was not sampled from the UK population but from the subset which had already been evaluated during the first assessment period, the covariate drift alerts may have been picking up these biases rather than shifts in overall population characteristics.

8.1 Further Work

The primary restrictions of this investigation is limiting the disease group to diabetes and allowing only an external exposome feature space. It is of interest to see if similar levels of model degradation are experienced by risk predictive models with alternative classification tasks, such as identifying participants at risk of developing cardiovascular disease, depression, or skin cancer, as well as allowing the predictive models to learn on blood work test results. In addition, the impact of batch size choice for the various detection methods has not been explored.

Currently there is no causal link between the detection of data drift and subsequent model degradation. We would like to further explore the potential for interpreting the impact of covariate drift on the models through comparison and analysis of normalized Shapley contributions. This leads to a natural extension of this thesis, an exploration of techniques for identification, interpretation, and correction of data drift driven performance degradation.

Chapter 9

Conclusion

In this thesis, several diabetes risk-predictive ML models were trained and validated using exposome data from the UKBB assessments between 2006 and 2010. The best performing architectures by average precision were Logistic Regression (0.592 ± 0.004) and XGBoost (0.594 ± 0.008). Each model was subsequently evaluated on a temporally distant cohort, in order to assess the traceability of the exposome-based models. Every model architecture experience a significant degradation in performance between the two time periods. Further analysis was run to explore possible driving forces of this degradation. Covariate data drift was detected within a subset of the UKBB exposome feature space. This drift was detected using both multivariate and univariate detection methods and was found in the Birth Year, BMI, Frequency of Tiredness, and Lack of Education features. Severe label drift was not detected. However, due to the lack of continuously collected data, a gradual shift in the label distribution could not be ruled out. Lastly, a model-aware concept drift detection method was applied by tracking temporal changes in normalized Shalpely contributions for model input features.

Due to a lack of causal connection between the detected drifts and the model degradation along with possible representation biases in the second reassessment cohort, further research is needed before definitive claims can be made about the susceptibility of exposome-based disease predictive models to data drift. However, it is clear that traceability of clinical models is a subject which requires careful monitoring to ensure that the benefits provided by AI/ML clinical solutions remain stable well past initial deployment.

Appendix A

UKBB Exposome Feature Space

Feature Category	Feature	
Physical	Waist Circumference	Hip Circumference
	Standing Height	BMI
	Weight	Body Fat Percent
	Whole Body Fat-Free Mass	Basal Metabolic Rate
	Hair Color	
Demographics	Year of Birth	Townsen Depreivation
Education	Level of Education	End of Education Age
Employment	Employment Status	Length Current Employment
	Length of Work Week	Work Home Distance
	Job Involves Standing	Job Involves Physical Work
	Job Involves Shift Work	
Lifestyle	Sleep Duration	Sleeplessness/Insomnia
	Daytime Dozing	Naps During Day
	Tobacco Smoking	Past Tobacco Smoking
	Frequency of Alcohol	Ever Taken Cannabis
	Injury Through Alcohol	Recommended Less Alcohol
	Alcohol Drinker Status	Use of Sun Protection
	Apparent Facial Age	Ease of Skin Tanning
	Time Outdoors (Summer)	Time Outdoors (Winter)
	Mobile Phone Use	Weekly Phone Usage
	Change in Phone Habits	Head Side Using Phone
	Plays Computer Games	
Environment	Traffic Intensity	Near Major Road
	Daytime Sound Level	Evening Sound Level

TABLE A.1: Physical, demographic, education, employment, lifestyle, and environmental features used from UKBB dataset.

Feature Category	Feature	
Diet	Cooked Vegetable Intake	Raw Vegetable Intake
	Fresh Fruit Intake	Dried Fruit Intake
	Oily Fish Intake	Non-oily Fish Intake
	Processed Meat Intake	Poultry Intake
	Beef Intake	Lamb Intake
	Pork Intake	Cheese Intake
	Milk Type	Bread Intake
	Bread Type	Cereal Intake
	Cereal Type	Added Salt
	Tea Intake	Coffee Intake
	Coffee Type	Varied Diet
	Spread Type	Water Intake
	Major Diet Change	Non-butter Spread Type
Diet (24 Hours)	Coffee Consumed	Tea Consumed
	Alcohol Consumed	Vitamin Supplements
Early Life	Childhood Sun Burns	Breastfed
	Age 10 Body Size	Age 10 Height
	Handedness	Adopted
	Multi-birth	Maternal Smoking Post-birth
	Medical Guardian	Molested
	Physically Abused	Felt Loved
	Felt Hated	
Trauma	Sexual Assault	Witnessed Violent Death
	Violent Crime	Life-threatening Accident
	Able to Pay Rent	Partner Sexual Assualt
	Partner Violence	Confiding Relationship
	Partner Belittlement	Disturbed by Past Trauma
	Upset by Past Trauma	Avoided Activities due to Past
Mental Health	Risk Taking	Depressed Mood
	Doctor Visit	Psychiatrist Visit
	Happiness	Week Long Depression
	Longest Depressed Period	Number Depressive Episodes
	Bipolar	Neuroticism Score
	Week Long Disinterest	Frequency Tired
	Frequency Tense	Frequency Disinterested

TABLE A.2: Dietary, early life, trauma related, and mental health features used from UKBB dataset.

Appendix B

Univariate Drift Analysis Figures

This section holds the univariate drift analysis figures generated for both the selected Logistic Regression and XGBoost model. Both the drift alert and the batched distribution plots are reported. Categorical and continuous features are separated as the distance metrics used for the drift alert detection method differed depending on the feature datatype.

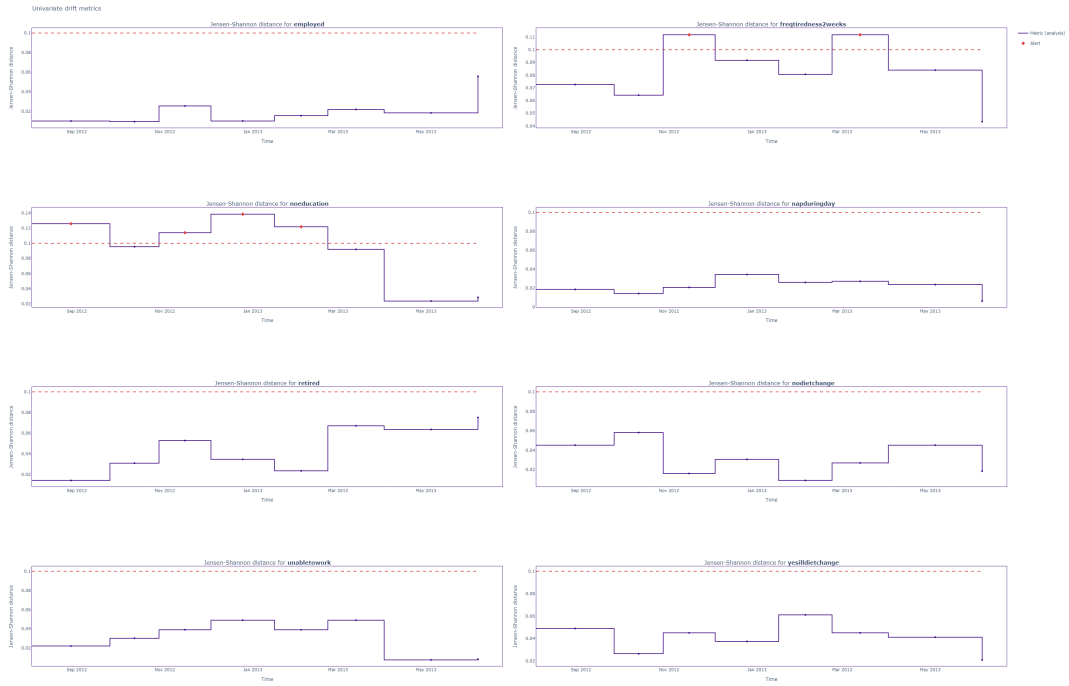


FIGURE B.1: Alert plots for categorical features in traceability cohort.

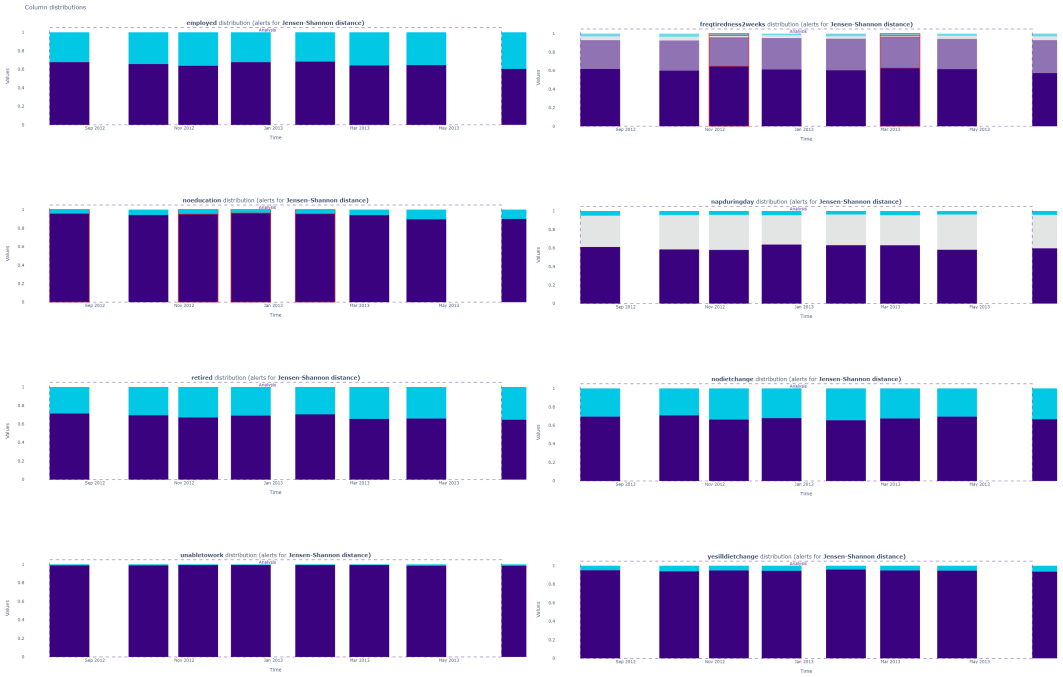


FIGURE B.2: Distribution plots for categorical features in traceability cohort. Batches which exceed the drift threshold are shaded in red.

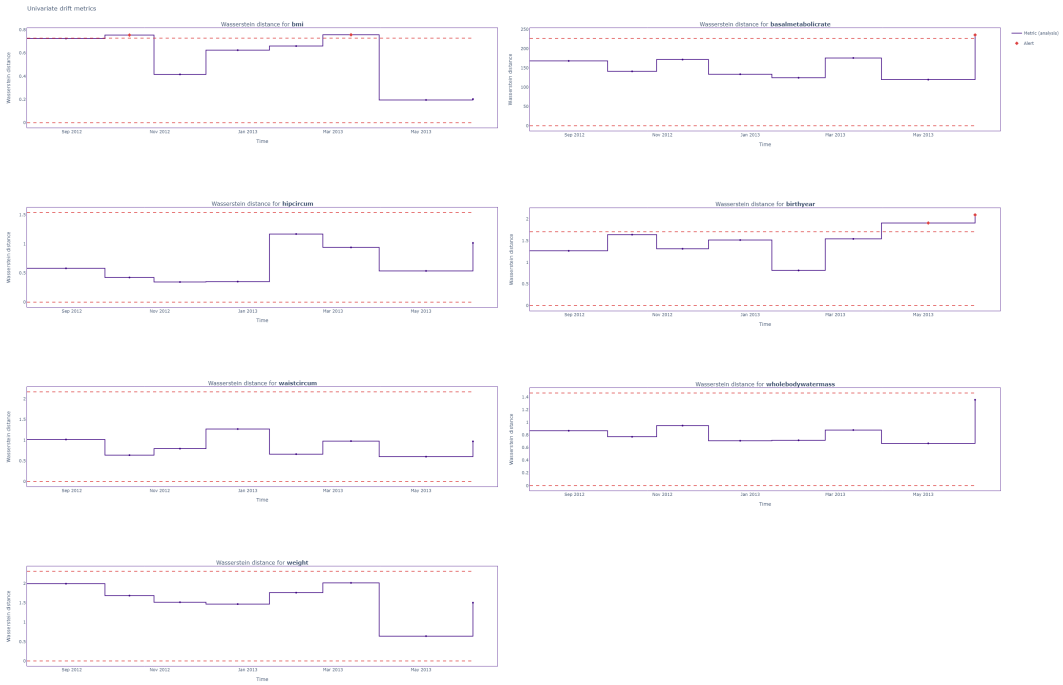


FIGURE B.3: Alert plots for continuous features in traceability cohort.

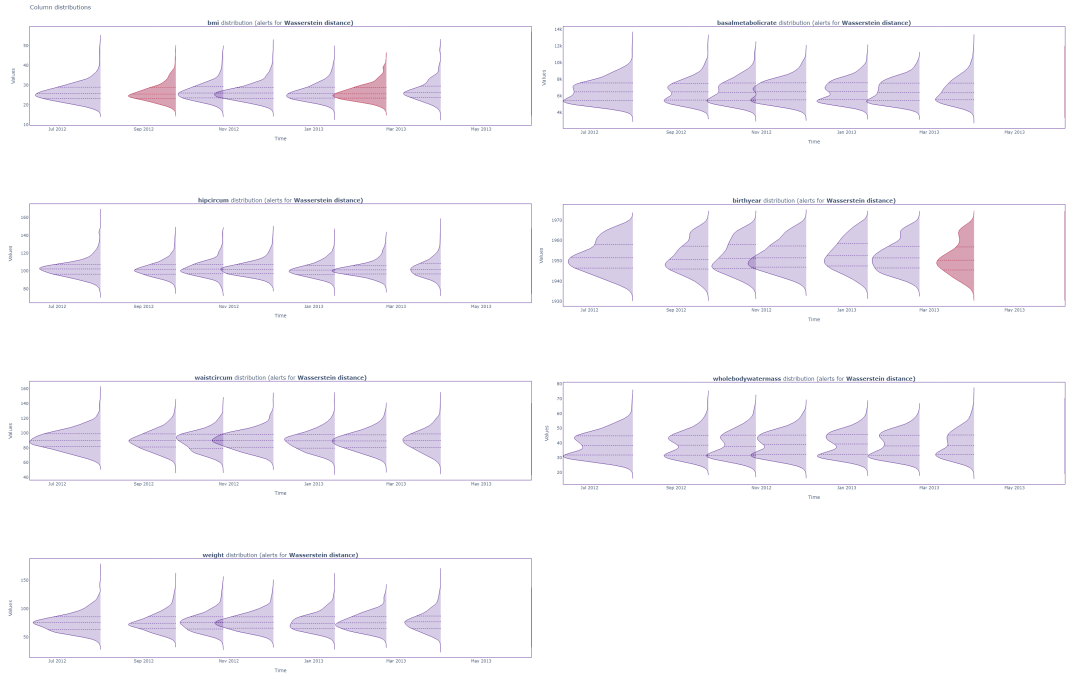


FIGURE B.4: Distribution plots for categorical features in traceability cohort. Batches which exceed the drift threshold are shaded in red.

Bibliography

Ackerman, Samuel et al. (2021). *Automatically detecting data drift in machine learning classifiers*. arXiv: 2111.05672 [cs.LG].

Bayram, Firas, Bestoun S Ahmed, and Andreas Kassler (2022). "From concept drift to model degradation: An overview on performance-aware drift detectors". In: *Knowledge-Based Systems*, p. 108632.

Beagley, Jessica et al. (2014). "Global estimates of undiagnosed diabetes in adults". In: *Diabetes Research and Clinical Practice* 103.2, pp. 150–160. ISSN: 0168-8227. DOI: https://doi.org/10.1016/j.diabres.2013.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0168822713003847.

Bhutani, Jaikrit and Sukriti Bhutani (2014). "Worldwide burden of diabetes". In: *Indian Journal of Endocrinology and Metabolism* 18.6, p. 868. DOI: 10.4103/2230-8210.141388.

Breiman, Leo (Oct. 2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

Buijsse, Brian et al. (May 2011). "Risk Assessment Tools for Identifying Individuals at Risk of Developing Type 2 Diabetes". In: *Epidemiologic Reviews* 33.1, pp. 46–62. ISSN: 0193-936X. DOI: 10.1093/epirev/mxq019. eprint: https://academic.oup.com/epirev/article-pdf/33/1/46/15546804/mxq019.pdf. URL: https://doi.org/10.1093/epirev/mxq019.

Char, Danton S., Michael D. Abràmoff, and Chris Feudtner (2020). "Identifying Ethical Considerations for Machine Learning Healthcare Applications". In: *The American Journal of Bioethics* 20.11. PMID: 33103967, pp. 7–17. DOI: 10.1080/15265161.2020.1819469. eprint: https://doi.org/10.1080/15265161.2020.1819469. URL: https://doi.org/10.1080/15265161.2020.1819469.

Chatterjee, Sudesna, Kamlesh Khunti, and Melanie J Davies (2017). "Type 2 diabetes". In: *The Lancet* 389.10085, pp. 2239–2251. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(17)30058-2. URL: https://www.sciencedirect.com/science/article/pii/S0140673617300582.

Chawla, Nitesh V et al. (2002). "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16, pp. 321–357.

Chen, Chao, Andy Liaw, Leo Breiman, et al. (2004). "Using random forest to learn imbalanced data". In: *University of California, Berkeley* 110.1-12, p. 24.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: KDD '16. San Francisco, California, USA: Association for Computing Machinery, 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

Christodoulou, Evangelia et al. (2019). "A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models". In: *Journal of Clinical Epidemiology* 110, pp. 12–22. ISSN: 0895-4356. DOI: https://doi.org/10.1016/j.jclinepi.2019.02.004. URL: https://www.sciencedirect.com/science/article/pii/S0895435618310813.

Cummings, Jesse, Elías Snorrason, and Jonas Mueller (2023). *Detecting Dataset Drift and Non-IID Sampling via k-Nearest Neighbors*. arXiv: 2305.15696 [cs.LG].

Duckworth, Christopher et al. (2021). "Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during COVID-19". In: *Scientific reports* 11.1, p. 23017.

Friedman, Jerome H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5, pp. 1189–1232. ISSN: 00905364. URL: http://www.jstor.org/stable/2699986 (visited on 06/15/2023).

Hoens, T. Ryan, Robi Polikar, and Nitesh V. Chawla (Apr. 2012). "Learning from streaming data with concept drift and imbalance: an overview". In: *Progress in Artificial Intelligence* 1.1, pp. 89–101. ISSN: 2192-6360. DOI: 10.1007/s13748-011-0008-0. URL: https://doi.org/10.1007/s13748-011-0008-0.

IBM (Oct. 2021). *IBM topic on logistic regression*. Accessed: 2023-06-14. URL: https://www.ibm.com/topics/logistic-regression.

Jaiswal, Varun, Anjli Negi, and Tarun Pal (2021). "A review on current advances in machine learning based diabetes prediction". In: *Primary Care Diabetes* 15.3, pp. 435–443. ISSN: 1751-9918. DOI: https://doi.org/10.1016/j.pcd.2021.02.005. URL: https://www.sciencedirect.com/science/article/pii/S175199182100019X.

Joshi, Ram D. and Chandra K. Dhakal (2021). "Predicting Type 2 Diabetes Using Logistic Regression and Machine Learning Approaches". In: *International Journal of Environmental Research and Public Health* 18.14. ISSN: 1660-4601. DOI: 10.3390/ijerph18147346. URL: https://www.mdpi.com/1660-4601/18/14/7346.

Kavakiotis, Ioannis et al. (2017). "Machine Learning and Data Mining Methods in Diabetes Research". In: *Computational and Structural Biotechnology Journal* 15, pp. 104–116. ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2016.12.005. URL: https://www.sciencedirect.com/science/article/pii/S2001037016300733.

Kotsiantis, Sotiris, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. (2006). "Handling imbalanced datasets: A review". In: *GESTS international transactions on computer science and engineering* 30.1, pp. 25–36.

Lekadir, Karim et al. (2021). "Future-ai: Guiding principles and consensus recommendations for trustworthy artificial intelligence in medical imaging". In: *arXiv preprint arXiv:2109.09658*.

MacKay, David JC (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Mani, S. et al. (2012). "Type 2 diabetes risk forecasting from EMR data using machine learning". In: *AMIA Annu Symp Proc* 2012, pp. 606–615.

McCradden, Melissa D et al. (2022). "A Research Ethics Framework for the Clinical Translation of Healthcare Machine Learning". In: *The American Journal of Bioethics* 22.5. PMID: 35048782, pp. 8–22. DOI: 10.1080/15265161.2021.2013977. eprint: https://doi.org/10.1080/15265161.2021.2013977. URL: https://doi.org/10.1080/15265161.2021.2013977.

Mitchell, T. (1997). "Mitchell: Machine Learning". In: *1997 Burr Ridge* 45.37, pp. 870–877.

NannyML (release 0.8.6) (Mar. 2023). https://github.com/NannyML/nannyml. NannyML, Belgium, OHL.

Nuyttens, Niels (2020). *Data Reconstruction with PCA*. URL: https://nannyml.readthedocs.io/en/stable/how_it_works/data_reconstruction.html#data-reconstruction-pca.

Panaretos, Victor M. and Yoav Zemel (2019). "Statistical Aspects of Wasserstein Distances". In: *Annual Review of Statistics and Its Application* 6.1, pp. 405–431. DOI:

10.1146/annurev-statistics-030718-104938. eprint: https://doi.org/10.1146/annurev-statistics-030718-104938. URL: https://doi.org/10.1146/annurev-statistics-030718-104938.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Ren, Shaoqing et al. (2015). "Global refinement of random forest". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, pp. 723–730.

Saeedi, Pouya et al. (2019). "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas". In: *Diabetes research and clinical practice* 157, p. 107843.

Saito, Takaya and Marc Rehmsmeier (Mar. 2015). "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets". In: *PLOS ONE* 10.3, pp. 1–21. DOI: 10.1371/journal.pone.0118432. URL: https://doi.org/10.1371/journal.pone.0118432.

Shrivastava, Abhinav Kumar et al. (2022). "Early Diabetes Prediction using Random Forest". In: *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1154–1159. DOI: 10.1109/ICESC54411.2022.9885683.

Su, Wanhua, Yan Yuan, and Mu Zhu (2013). *Threshold-free Evaluation of Medical Tests for Classification and Prediction: Average Precision versus Area Under the ROC Curve*. arXiv: 1310.5103 [stat.ME].

Uddin, Shahadat et al. (Dec. 2019). "Comparing different supervised machine learning algorithms for disease prediction". In: *BMC Medical Informatics and Decision Making* 19.1, p. 281. ISSN: 1472-6947. DOI: 10.1186/s12911-019-1004-8. URL: https://doi.org/10.1186/s12911-019-1004-8.