STAT 561 Project: Comparing Sampling Methodologies in Transfer Learning on Imbalanced Data

Andrei Afilipoaei, Shreya Kala, Jainish Mehta, Canzhu Song, Linglong Kong

December 10, 2024

Abstract

In this work, we investigate various sampling techniques to reduce Type II (False Negative) error in predicting Diabetes and Heart Disease risk using DNN Transfer Learning. For this, we implement Transfer Learning algorithms to train a model on a source dataset with corresponding source task, then apply the pre-trained model to a target task on target data. However, imbalanced source and target data severely impaired classification accuracy in Transfer Learning, so we propose and investigate the effect of various sampling methods (categorized in Random Undersampling and Random Oversampling methods) in reducing data imbalance. Firstly, we provide theoretical proofs showing how these sampling methods reduce Type II error; we then apply DNN Transfer Learning to a large-scale Diabetes risk factor dataset to verify our theoretical results and identify the optimal sampling technique for reducing False Negative error in Transfer Learning. In so doing, we observe that SRS Undersampling performs best in predicting both diabetes and heart disease risk in low-education demographics.

Contributions

- **Andrei Afilipoaei:** Project conceptualization, research and literature review, dataset collection, theoretical results and proofs, DNN coding, multi-layer DNN and fine-tuning, application of Random Undersampling and Oversampling techniques, Transductive TL and fine-tuning results compilation, project writing, presentation writing.
- **Shreya Kala:** DNN coding and fine-tuning, application of Random Undersampling and Oversampling techniques, research, brainstorming.
- **Jainish Mehta:** DNN coding, application of Random Undersampling and Oversampling techniques, results compilation.
- **Canzhu Song:** Research and project editing, presentation editing.

Contents

1 Introduction

The past decade has brought enormous changes in the realm of data science and data analytics, as advancing technology has allowed for the creation of increasingly large-scale and complex datasets [1]. This new-found access to enormous quantities of data has already seen extensive use across a wide variety of topics [2]. A major development in this regard is the wide-spread propagation of Machine Learning techniques in predicting future trends based upon previously-observed patterns, although the effectiveness of standard Machine Learning degrades considerably if there is a difference between the training and testing data [1, 2]; this has effectively limited Machine Learning to situations in which training and testing/prediction data have the same distributions and input feature spaces [2, 3]. This problem is further compounded by the tremendous cost of data collection [4], which severely limits the usefulness of Machine learning analysis in a variety of topics.

To overcome this significant weakness in traditional machine learning, a new technique has been proposed, called *Transfer Learning* [2]. The principle underlying Transfer Learning is inspired by human learning processes: more efficiently learning to perform a new task by *transferring* knowledge from a related task [2]. In intuitive terms, Transfer Learning occurs in musical instruments: a violin player may find it easier to learn to play the piano than someone with no musical experience [3]. In computing, Transfer learning seeks to use the information from an existing topic (termed the *source domain*) to more effectively or efficiently learn a new topic (termed the *target domain*) [3]. Transfer learning has proved extremely useful in helping to improve machine learning efficiency by cutting down on the necessary model training time [4]; namely, transfer learning allows analysts to avoid needing to build, train, and test a new model for each new task [4] by instead training a single model and then simply fine-tuning it to each task as needed. It can also be used to train a model on simulation data of natural processes and then fine-tune it to a smaller sample of real-world data [4]; this allows researchers to achieve necessary accuracy without needing to collect large datasets in situations in which doing so would be prohibitively expensive or even hazardous to humans and the environment [4].

In this project, we will investigate the effectiveness of Transfer Learning Deep Neural Networks in predicting diabetes and heart disease risk in low-education demographics from the large-scale "CDC Diabetes Health Indicators" dataset in the UCI Machine Learning repository [5], adapted from the 2015 Behavioral Risk Factor Surveillance System (BRFSS) [6]. However, since this dataset is highly imbalanced between the incidence rates of diabetes in the population (as is common for many datasets in medicine), both traditional Machine Learning and (by extension) Transfer Learning struggle to accurately predict disease risk [7]. As such, in this work we will investigate the effectiveness of various sampling techniques in improving the predictive accuracy of Transfer Learning DNN (as measured by False Negative Error) in quantifying diabetes risk in underrepresented demographics.

While extensive work has been done on sampling techniques from the standpoint of traditional Machine Learning ([1, 8]), literature on the topic of sampling in Transfer Learning (including on imbalanced datasets) has been limited; it is the aim of this article to further develop this area of study. For this reason, the primary study questions we will investigate in this work are, respectively, *how can sampling techniques help reduce False Negative error in disease risk factor prediction on an imbalanced dataset?* and *which sampling techniques are most effective, in terms of both training time and reducing False Negative Error?*

To investigate this, in Section 2 we will provide some background and definitions with reference to Transfer Learning, various sampling techniques, and the methods of resolving the data imbalance issue. Following this, in Section 3, we provide an outline of the sample survey methodology of the Behavioral Risk Factor Surveillance System [6], as well as potential coverage errors in this survey. In Section 4, we propose several sampling techniques to improve Transfer Learning classification performance on imbalanced datasets, and provide theoretical proofs demonstrating how these sampling techniques decrease Type II (False Negative) error on a Bayesian classifier. In Section 5, we then apply a Deep Neural Network Transfer Learning algorithm to classifying diabetes risk subject to various sampling methods, and compare the performance of these methods

in terms of both predictive accuracy and training time. In Section 6, we apply fine-tuning to predict heart disease risk in low-education demographics using a pre-trained model on high-education diabetes risk, and compare the effectiveness of the different under/oversampling techniques on the target training set in reducing False Negative Error. Finally, in Section 7, we summarize our findings and discuss potential areas of further development.

2 Background & Literature Review

2.1 Transfer Learning

Recent years have brought the development of large-scale and complex datasets, which have in turn enabled the wide-spread proliferation of new data analysis and decision-making techniques including Machine Learning [1]; this new-found access to enormous quantities of data has already seen extensive use in a wide variety of applications [2]. However, the effectiveness of traditional Machine Learning relies on keeping the training and testing/prediction data on the same distributions and input feature spaces [2, 3]. Given the high cost of data collection [4], the new technique of *Transfer Learning* has been proposed to learn from related tasks in order to make Machine Learning more effective and efficient [2].

By its design, Transfer learning is of great use in situations in which large-scale target datasets are unavailable [2]; causes of such limited datasets include the necessary data being inaccessible, inordinately expensive or difficult to collect, or exceedingly rare [2]. As Ayana et al. (2024) point out, Transfer Learning methodologies have seen extensive medical use in generalizing diagnostic processes and overcoming scarcity in data [9]. In cases in which diagnosis is needed for very rare medical conditions (which by nature do not have large datasets) or in which different medical conditions have varying weights within the sample (class imbalance), Transfer learning has demonstrated improved diagnosis capabilities as well as an improvement in model training efficiency [9]. Other (non-medical) uses of transfer learning have included methodologies to classify and distinguish

between a variety of different objects, including cars and bicycles [4]. Another proposed topic of development of Transfer Learning, according to Hosna et al. (2022), is in being able to transfer source-domain data (and corresponding ML-trained knowledge) from one organization to another without compromising confidentiality [4].

Transfer learning is not without its set of drawbacks, however. The greatest danger when attempting to transfer a pre-trained model to a new task is that the model performance can actually be *worse* than if no pre-training had been done at all [2, 4]; in other words, the pre-existing data can actually have a *negative* effect on the performance and accuracy of the target Machine Learning task [2]. This well-known problem is called *negative transfer* [2, 4], and as Hosna et al. (2022) point out, has remained a significant unresolved issue for Transfer Learning to date [4]. As such, recent years have seen extensive work on maximizing the effectiveness and efficiency of Transfer Learning, while simultaneously limiting the danger of negative transfer in the process [4]. In all, Transfer Learning has been shown to be most effective if three important conditions are met: (1) the source and target tasks are similar to one another; (2) the distributions of the source and target datasets are relatively close to each other; and (3) both tasks can be accomplished with comparable learning models [10].

2.1.1 Mathematical Background & Terminology

According to Weiss et al. (2016) [2], in the context of transfer learning a *Domain* $\mathscr{D} = \{\chi, P(X)\}$ consists of a feature space χ for which a particular learning example $X = (x_1, x_2, ..., x_n) \in \chi$ and a marginal probability distribution $P(X)$ [2]. In this case, the learning example X consists of n feature vectors (instances), term x_i denotes the i^{th} feature vector in the learning example X, and χ is the space of all (possible) feature vectors [2]. Furthermore, given such a domain $\mathscr{D}$, Weiss et al. (2016) further define a task $\mathscr{T} = \{Y, f(\cdot)\}$ as consisting of a label space Y and a corresponding predictive function $f(\cdot)$ that has been trained on the pairs of feature vectors and labels $\{x_i, y_i\}$ corresponding to the learning example X and label space, respectively (i.e. for $x_i \in X$ and $y_i \in Y$).

In that case, Weiss et al. (2016) [2] define the *source domain data* D_S as the

pairs of feature vectors and labels from the source feature space χ_S and source label space Y_S, respectively; as such, the source domain data can be written as $D_S = \{(x_{S1}, y_{S1}), (x_{S2}, y_{S2}), ..., (x_{Sn}, y_{Sn})\},\ x_{Si} \in \chi_S,\ y_{SI} \in Y_S$. In similar fashion, the *target domain data* is denoted as $D_T = \{(x_{T1}, y_{T1}), (x_{T2}, y_{T2}), ..., (x_{Tn}, y_{Tn})\}$ for $x_{Ti} \in \chi_T,\ y_{Ti} \in Y_T$ as the pairs of feature vectors and labels from the target feature space χ_T and target label space Y_T, respectively [2]. Based on the indications of Weiss et al. (2016), the source task is defined as $\mathscr{T}_S = \{Y_S, f_S(\cdot)\}$ for the source predictive function $f_S(\cdot)$ trained on the source domain data, and the target task is defined as $\mathscr{T}_T = \{Y_T, f_T(\cdot)\}$ for the target predictive function $f_T(\cdot)$ trained on the target domain data [2].

Using these definitions, Weiss et al. (2016) [2] formally define the process of transfer learning as follows: *given a source and target domain $\mathscr{D}_S$ and $\mathscr{D}_T$, respectively, and given corresponding source and target tasks $\mathscr{T}_S$ and $\mathscr{T}_T$, respectively,* Transfer Learning *seeks to improve the target predictive function $f_T(\cdot)$ using related information from $\mathscr{D}_S$ and $\mathscr{T}_S$ for which either the source domain or source task does not match the target domain and/or target task (i.e. $\mathscr{D}_S \neq \mathscr{D}_T$ and/or $\mathscr{T}_S \neq \mathscr{T}_T$)* [2]. They go on to point out, for the domain definition outlined above (of the form $\mathscr{D} = \{\chi, P(X)\}$) that a difference between the source and target domain $\mathscr{D}_S \neq \mathscr{D}_T$ may imply either $\chi_S \neq \chi_T$ (the source and target feature spaces are different) or $P(X_S) \neq P(X_T)$ (the source and target marginal distributions of the input spaces differ) [2]. Meanwhile, a difference between the source and target tasks $\mathscr{T}_S \neq \mathscr{T}_T$ may imply either a difference in the source vs. target label/class spaces ($Y_S \neq Y_T$) or a difference between the source and target conditional probability distributions ($P(Y_S|X_S) \neq P(Y_T|X_T)$) [2].

Furthermore, Weiss et al. (2016) formally define *negative transfer* as the situation in which a target predictive function $f_{T1}(\cdot)$ trained only on the target domain $\mathscr{D}_T$ actually performs *better* than a target predictive function $f_{T2}(\cdot)$ trained on a combination of $\mathscr{D}_S$ and $\mathscr{D}_T$ combined [2].

2.1.2 Types of Transfer Learning

There are three general categories in Transfer learning, according to Hosna et al. (2022): *Inductive TL*, *Transductive TL*, and *Unsupervised TL* [4]. It should be noted, however, that this list is not exhaustive [10].

Inductive TL: Here, the target domain is the same as the source domain, but the target task to which the Transfer Learning process is applied differs from the source task [4]. Inductive TL processes also fall into two general categories: *Multi-task learning* and *Self-taught learning* [4]. In *Multi-task learning*, the source and target domain are identical, but the aim of TL is to maximize the performance of multiple tasks beyond the scope of the original pre-trained model [4, 10]. Meanwhile, in *Self-taught learning*, the labelling of source data is unavailable, and the source and target domain may not be equal [4]. Inductive TL is often considered a variant of supervised learning [10].

Transductive TL: The target task matches the source task, but the target and source domains are different [4]; for instance, it is often the case in Transductive TL that the source data has labelling information while the target data does not [10]. One major example of Transductive TL mentioned is *domain adaptation*, in which performing a given task on the labelled data from one distribution is used to transfer knowledge to applying the same task to a different distribution [10].

Unsupervised TL: Neither the target nor source domains have labelled information, and Transfer Learning is used for unsupervised learning processes such as dimension reduction and clustering [4]. A major area of focus for Unsupervised TL is detecting fraud through the analysis of generalized patterns from financial transaction data [10].

Meanwhile, Weiss et al. (2016) further discuss two processes in Transfer learning: *Homogeneous TL* and *Heterogenous TL* [2].

Homogeneous TL: Using the formal definitions outlined by Weiss et al. (2016), Homogeneous Transfer Learning takes place when the source and target feature spaces match ($\chi_S = \chi_T$) [2]. Homogeneous transfer learning is used extensively in machine learning tasks with large-scale datasets in related domains [2]; in such a case, Homogeneous TL is used to construct a predictive function for the target domain so long as the

feature spaces for the source and target domains match [2].

Heterogenous TL: Again using the formal definitions outlined by Weiss et al. (2016), Heterogeneous Transfer Learning takes place when the source and target feature spaces differ ($\chi_S \neq \chi_T$) [2]. In a big data environment, Heterogeneous TL can be used to bridge the gap between a large-scale database and a target database that are both from the same target domain but with different feature spaces, and it can help to build a predictive model in that target domain [2]. The uses and applications of Heterogenous TL are extensive, ranging from image recognition to classification tasks in single- and multiple-language texts, software, human activity, and drug efficacy [2], although Weiss et al. (2016) indicate that this area of Transfer Learning is relatively recent [2].

2.2 Imbalanced Datasets & Sampling

In the realm of Machine Learning classification, a major problem often encountered is the issue of *imbalanced data* [7]. Namely, current Deep Neural Network algorithms yield excellent performance in data-balanced cases (in which the different classes being studied in the dataset are of approximately equal size), but tend to perform much worse in cases where the data is heavily weighted (imbalanced) toward one or more classes in the data [7]. This is because imbalanced data leads the Neural Network to learn more predominantly from the majority class to the detriment of predicting the minority class [7]. This notable problem with Machine Learning is especially significant in areas such as monitoring, medicine, and industry [7], in which one class (such as disease positivity) is heavily outnumbered in the dataset by another class (disease negativity). In their work, Liu et al. (2023) proposed the implementation of Transfer Learning and Active Sampling for the purpose of fine-tuning the performance of traditional Deep Neural Networks on imbalanced data [7]; this reinforces the importance of sampling techniques in remedying the problems caused by imbalanced datasets in Machine Learning.

According to Wongvorachan et al. (2023), there are several different methods of dealing with highly imbalanced datasets [8]; these methods generally fall into two different overall groups: *undersampling* and *oversampling* [8].

Undersampling methods, according to Wongvorachan et al. (2023), strive to sample elements from the predominant (majority) class to obtain a sample that contains equal numbers in both majority and minority classes [8], such that the problem of data imbalance is no longer an issue. A notable (albeit simple) method often employed for the purpose of undersampling is called *Random Undersampling* (RUS), which conducts random sampling (without or with replacement) from the majority class to create a sample of equal size to the minority class [8]. A problem commonly encountered in undersampling, Wongvorachan et al. (2023) further point out, is that reducing the number of elements in the majority class risks losing valuable information that would be used in prediction, although it helps to reduce training time in the dataset as well [8].

On the other hand, Oversampling methods strive to either resample or generate new (artificial) elements within the minority class, such that the overall minority class population increases to match that of the majority class [8]. A simple oversampling technique, termed *Random Oversampling* (ROS), involves resampling existing datapoints from the minority class to obtain duplicated datapoints for use in balancing the dataset [8]; however, as Wongvorachan et al. (2023) warn, this ROS technique increases the risk of overfitting because of the large number of resultant duplicated datapoints [8]. As such, a second common oversampling method proposed (heavily based on ROS methodology) is the Synthetic Minority Oversampling Technique (SMOTE), which functions by generating new (artificial) elements in the minority class with focus on close neighbors of existing minority elements [8]. Some notable problems of oversampling techniques in general (including both ROS and SMOTE, among others) are that (1) they increase the risk of overfitting; (2) that some synthetic oversampling techniques (such as SMOTE) perform poorly on higher-dimensional data; and (3) they increase the training time of the model because of the larger resultant dataset [8].

There are several Random selection techniques that are pertinent to the process of Under/Oversampling, including *Simple Random Sampling*, *Stratified Sampling*, *Cluster Sampling*, and *Systematic Sampling* [1, 11].

Simple Random Sampling (SRS) is among the most commonly-employed random

selection techniques due to its ease of use [1]. In particular, SRS techniques work well in populations or datasets for which there are no significant clusters or outliers in the data, allowing for a straightforward selection process of datapoints from the overall dataset [1]; most SRS techniques require that the overall population/dataset size N be known prior to sampling, although SRS techniques exist to bypass this limitation (including *Random sampling with Reservoir*, *Acceptance/Rejection Sampling*, and *Weighted Sampling*, among others) [1].

Stratified Sampling, on the other hand, divides the overall dataset (population) into multiple nonoverlapping groups termed *strata*, and then conducts independent sampling from each of the strata [1]. Besides needing to specify the overall sample size, Stratified Sampling also requires a defined allocation method across the different strata [11]; three common allocation methods for this purpose include *Optimal Allocation*, *Neyman Allocation*, and *Proportional Allocation*, respectively [1]. A major drawback of stratified sampling is that it is much more time- and cost-intensive to perform in the creation of the strata [1], although the break-up into nonoverlapping but internally homogeneous groups can make it more convenient and effective [11].

Cluster sampling functions by grouping the population into nonoverlapping "clusters", after which Simple Random Sampling is used to select certain clusters from the population [1]. Cluster sampling is often employed in situations in which it is prohibitively expensive or difficult, or even impossible, to collect an accurate list (sample frame) of all units in the population, but relatively straightforward to collect a list of different clusters [11]. Another variant of this sampling technique, called *two-stage sampling*, selects a given number of clusters using a sampling technique and then selects a given sample from each of those clusters [12, 13]. Cluster sampling, however, can exhibit problems in cases in which the clusters are themselves highly imbalanced, although recent work has studied how resampling techniques (including bootstrapping) can be used to overcome these hurdles [1].

Finally, **Systematic Sampling** uses a defined order of the population/dataset, selects a random starting point from the first k terms, and takes every subsequent k^{th} term

in the dataset [11]. In situations in which the population size N is not an integer multiple of k, different systematic samples may vary in size by 1 unit [11]. By its functioning, systematic sampling resembles cluster sampling by breaking the population into k clusters and conducting one-stage cluster sampling with only one cluster [11]. Systematic sampling is useful because it can be even easier to select a sample than the Simple Random Sampling selection technique, and since a systematic sample is more evenly spread across the dataset, it is more precise than simply performing stratified sampling with one unit per stratum [11].

3 Dataset Sampling Survey Methodology & Error

In this project, we will study the effectiveness of various sampling methodologies on improving diabetes and heart disease risk prediction from the "CDC Diabetes Health Indicators" dataset in the UCI Machine Learning repository [5]. This dataset contains a total of 22 different features (including Age, Sex, Mental Health, Smoking, Alcohol Consumption, and Education Level, among others) [5] as well as a binary classification label with 1 denoting either prediabetes or diabetes and 0 denoting no diabetes [14]. The dataset comprises of over 250,000 survey responses to the 2015 Behavioral Risk Factor Surveillance System (BRFSS) conducted by the Centres for Disease Control and Prevention in the United States [6]. The BRFSS is a telephone-based survey that collects data from all 50 states and 3 US territories, comprising of over 400,000 interviews conducted yearly [6].

Despite an extensive and thorough survey methodology, the BRFSS has experienced difficulties with data collection and survey response, especially in recent decades [15, 16]; this likely arises from many factors including access to landlines that could bring significant coverage error [15]. Notably, Peytchev et al. (2011) mention that nearly 95% of the US adult population had a landline in 1996 compared to only 87% in 2006 [15]; at the same time, there was a significant decline in response rates to all types of telephone surveys, with the BRFSS suffering a 19% drop in response rate over that period [15].

To deal with this shifting trend away from landline telephones among the population, in 2011 the BRFSS survey methodology was changed to incorporate both new weighting methods and the use of cellular phones [16]. Despite this, however, the response rate has remained (relatively) low, with the 2023 BRFSS registering only a 44.7% response rate overall with a mean completed interview rate of only 1.8% for landlines and 3.4% for cell phones [16].

The sample survey data collected by the BRFSS is weighted using both *design weighting* (using stratified sampling methods) and *raking* (iterating the fitting proportion) to attempt to minimize demographic differences from the sample selected to the group which it represents [17]. For the purpose of calculating the design weighting of the survey, the BRFSS uses stratification by geographic location and calculates design weight w_i by the equation

$$w_i = W_{str} * \frac{N_{adult}}{N_{phones}}$$

Where W_{str} is the stratum weight, N_{adult} is the number of adults in a household, and N_{phones} is the number of landlines in that household [17]; as the CDC points out in its BRFSS documentation, cell phone interviews are treated as $N_{adult} = 1$ and $N_{phones} = 1$ [17]. As such, the weighting methodology of the BRFSS displays heavy use of stratified sampling methodologies but also some *cluster sampling* methodologies in analyzing household units as opposed to individuals [17].

As Peytchev et al. (2011) point out, the crux of any probability-based survey is the ability to sample and collect data from any and all members of the target population [15], although this, in practice, is not feasible. In particular, Peytchev et al. (2011) warn that the inability to contact or receive responses from certain members of the population causes their inclusion probability to become 0 or unknown, respectively [15]. This is especially problematic for a telephone survey such as the BRFSS, since coverage error [15] is likely to be introduced in under-representing individuals from poorer demographics, since these are more likely to not have access to either a landline or a cell phone and therefore not be included on the sampling frame. Therefore, considering the declining coverage of landlines [15] and the problem of coverage error in the population, it is likely

that the BRFSS will under-represent poorer demographics that could significantly bias the overall risk factor prediction.

4 Theoretical Results

To begin this analysis, we wish to determine which random sampling techniques (if any) help to improve efficiency and predictive accuracy in classifying diabetes risk from the UCI dataset [5]. Before implementing Transfer Learning on the Diabetes dataset, it is first instructive to prove how under/oversampling techniques impact the Type II (False Negative) error rate on imbalanced data. Because the BRFSS (and hence UCI dataset) records risk factors for disease [5, 6], it is preferable for diagnostic purposes to generate a conservative classifier with a reduced Type II (false negative) error, even if that brings an increase in Type 1 (False Positive) error; this would help to reduce the risk of diabetic or prediabetic conditions going undetected despite the prevalence of certain risk factors.

Drawing from the terminology outlined by Zadrozny (2004), the data for a binary classification problem takes the form (x, y) where x denotes the feature vector and y denotes the corresponding binary label [18]. In that case, sampling from the overall dataset/population introduces another data component $\mathscr{I}$ denoting the binary sample selection indicator function I_i for each term in the dataset. As such, the binary classification data now takes the form (x, y, I) (in similar form to that given by Zadrozny (2004) [18]). Zadrozny (2004) further lists four possible cases relating to the independence (or lack thereof) of the sampling method with regards to the values of x and y: namely, (1) that the sampling I is altogether independent of the values of x and y; (2) that the sampling is independent of y *given* x (i.e. $P(I|x, y) = P(I|x)$); (3) that the sampling is independent of x *given* y (i.e. $P(I|x, y) = P(I|y)$); and (4) that the sampling does not fulfill any independence assumption with regards to x and y [18]. It should be noted that this fourth case requires knowing a feature vector x_s for the sample to yield any information on mapping the feature vector x into the corresponding label y [18].

Since there is a strong relationship between physical health, Body Mass Index, in-

cidence of stroke, and heart disease [19, 20, 21], all of which are features in the UCI Diabetes dataset [5], the Naive Bayes assumption outlined by Zadrozny (2004) [18] is not valid for Diabetes risk factors. Therefore, we will instead analyze the Bayes classifier outlined by Zadrozny (2004) [18] of the form

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{1}$$

4.1 Classifier Conditional Probabilities

Since the sample consists only of those data points/elements for which the sampling indicator function $I_i = 1$, the sampling classifier conditional probability takes the form

$$P(y|x, I = 1) = \frac{P(x|y, I = 1)P(y|I = 1)}{P(x|I = 1)}\,. \tag{2}$$

As such, we investigate the first three cases outlined by Zadrozny (2004): (1) the sampling I is altogether independent of the values of x and y; (2) the sampling is independent of y *given* x (i.e. $P(I|x, y) = P(I|y)$); (3) the sampling is independent of x *given* y (i.e. $P(I|x, y) = P(I|x)$) [18]. In Cases 1 and 2, it can be observed that

$$P(y|x, I = 1) = \frac{P(x|y, I = 1)P(y|I = 1)}{P(x|I = 1)} = \frac{P(x|y)P(y)}{P(x)} = P(y|x)\,.$$

Meanwhile, Case 3 (for which $P(I|x, y) = P(I|x)$) yields

$$P(y|x, I = 1) = P(y|x) * \frac{P(I = 1|y)}{P(I = 1|x)} \tag{3}$$

The proofs of these properties are given in Appendix A at the end of the article.

As such, it can be observed that Case 3 is susceptible to sampling bias in estimating the classifier conditional probability, while Cases 1 and 2 are not; this confirms the conclusions drawn by Zadrozny (2004) [18]. However, although Zadrozny (2004) viewed this as unfavorable for the purpose of accurate sampling classification [18], this sampling 'bias' will serve as the primary driver of performance improvement in Random

Under/Oversampling on imbalanced datasets, as will be shown subsequently.

4.2 Random Undersampling (RUS)

According to Wongvorachan et al. (2023), there are two general techniques for dealing with highly imbalanced data: *undersampling* and *oversampling* [8]. In this section, we will analyze the undersampling technique known as *Random Undersampling* (RUS), which uses random sampling (without or with replacement) from the majority class to create a sample of equal size to the minority class [8]. As such, we will investigate the effectiveness of several different sampling techniques in performing Random Undersampling. Since the Transfer Learning process we will be conducting involves binary classification as the target task, the following results apply in equal measure to Transfer Learning as to traditional Machine Learning classification.

4.2.1 Simple Random Sampling

First, we investigate the effectiveness of performing Simple Random Sampling *without replacement* to select a balanced subset of the overall dataset. For this, we take N_0 to denote the total number of majority elements in the overall dataset/population, and N_1 to denote the total number of minority elements in the overall dataset (subject to the condition $N_1 < N_0$). Therefore, for the majority class undersampling, we define an indicator function I_{0i} to denote whether the feature vector x_{0i} will be sampled from the majority class, with $I_{0i} = 1$ if x_{0i} is sampled and $I_{0i} = 0$ otherwise. As such, since we will be conducting SRS to collect a sample of size N_1 from the majority class of size N_0, the expected value of this indicator function, denoting the probability of sampling x_{0i} from the majority class, is $E(I_{0i}) = \frac{N_1}{N_0}$.

On the other hand, all elements x_{1i} in the minority class are kept in the undersampled dataset, so there is no randomness involved in the minority class and $I_{1i} = 1$. Hence, we observe that the probability of sampling an element using SRS Random Undersampling

given the label y for that output is

$$P(I = 1|y) = \begin{cases} \frac{N_1}{N_0} \text{ if } y = 0 \\ 1 \text{ if } y = 1 \end{cases} \tag{4}$$

Therefore, we determine that SRS Random Undersampling yields a sampling probability that is conditional on the label y. However, since this SRS is independent of the value of feature vector x within each class, we conclude that the probability of sampling a term from the dataset is independent of x given y, thereby fulfilling the condition $P(I|x, y) = P(I|y)$.

Therefore, using Equation 3, we determine that the classifier conditional probability using SRS undersampling is

$$P(y = 1|x, I = 1) = P(y = 1|x) * \frac{P(I = 1|y = 1)}{P(I = 1|x)} \geq P(y = 1|x)\,. \tag{5}$$

The proof of this property is given in Appendix B at the end of the article.

In fact, only feature vectors which reside solely in the minority class will fulfil the conditional probability $P(I = 1|x) = 1$; all other feature vectors (including those which appear only in the majority class as well as those that appear in both classes) instead fulfill $P(I = 1|x) < 1$, in which case Equation 5 yields $P(y = 1|x, I = 1) > P(y = 1|x)$. Hence, the conditional probability of predicting $y = 1$ (diabetes positivity) given a feature vector x is higher in the SRS undersampled dataset than in the full (imbalanced) dataset, thereby helping to reduce Type II (False Negative) error in Transfer Learning classification.

However, for feature vectors x_i for which it is highly improbable for $y = 1$, the conditional probability $P(I = 1|x)$ will be much closer to $\frac{N_1}{N_0} << 1$, in which case $P(y = 1|x, I = 1) >> P(y = 1|x)$. This indicates that the problem of False Positive Error will also increase significantly using SRS Undersampling.

4.2.2 SRS With Replacement

Now, we turn our attention to examining the Random Undersampling technique using Simple Random Sampling *with replacement*. In this case, we again select a sample of size N_1 from the majority class of size N_0. As such, by the indications of Cochran (1977) [11], we replace the sampling indicator function I_i from the previous section with a *times* function t_i that counts the *number* of times that a particular feature vector x_i is selected. Since Random Undersampling takes the full minority class (corresponding to $y = 1$), the times function for the minority class is $t_{1i} = 1$.

Therefore, the sample classifier conditional probability takes the form

$$P(y = y'|x, t \geq 1) = P(y = y'|x) * \frac{P(t \geq 1|y = y')}{P(t \geq 1|x)} \geq P(y = y'|x) \tag{6}$$

The proof of this property is given in Appendix C at the end of the article.

Furthermore, feature vectors x that are also located in the majority class will fulfil $P(y = y'|x, t \geq 1) > P(y = y'|x)$. Therefore, we conclude that Undersampling using Simple Random Sampling with Replacement increases the conditional probability of predicting $y = 1$ (diabetes positivity) given a feature vector x, so SRSwR Undersampling reduces the problem of Type II (False Negative) error in Transfer Learning classification.

4.2.3 Systematic Undersampling

Now, we examine the possibility of using systematic sampling methods to perform Random Undersampling of the imbalanced dataset. As outlined by Cochran (1977), a 1-in-k systematic sample involves ordering the dataset, randomly choosing a starting point between 1 and k, and then taking every subsequent k^{th} element [11]. This yields a sample of size $n = \frac{N}{k}$ (assuming the population size N is an integer multiple of k), although the final sample size may vary by one unit if the population size is not an integer multiple of k [11].

For convenience, we assume that the majority class population size N_0 is an integer multiple of the minority class population size N_1, such that we take the systematic

sampling interval to be $k = \frac{N_0}{N_1}$. As such, we again take the indicator function I_{ij} to denote whether feature vector x_{ij} from the i^{th} class is included in the sample. For the minority class, we have $I_{1j} = 1, \forall j \in \{1, 2, ..., N_1\}$, as all elements from the minority class are sampled exactly once. On the other hand, for the majority class, we have a random indicator function I_{0j} denoting whether the element from the majority class is included in the undersampled dataset; based on the indications of Cochran (1977) [11], the expected value of this indicator function is $E[I_{0i}] = \frac{1}{k} = \frac{N_1}{N_0}$, which is the same as the expected value in the Simple Random Sample in Section 4.2.1.

Therefore, using the same rationale as in Section 4.2.1, we conclude that

$$P(y = 1|x, I = 1) \geq P(y = 1|x)$$

Where, with the exception of feature vectors which only appear in the minority class, the condition is actually $P(y = 1|x, I = 1) > P(y = 1|x)$.

Therefore, we determine that the conditional probability of predicting $y = 1$ (diabetes positivity) given a feature vector x is higher in the Systematic Undersampled dataset than in the full (imbalanced) dataset, so Systematic Undersampling reduces Type II (False Negative) error in Transfer Learning classification.

This does not mean that overall performance will be equal for Systematic Undersampling as for SRS undersampling, however. Namely, depending on how the dataset is ordered, Systematic Undersampling might over- or under-represent a certain feature vector in the undersampled majority class and thereby result in a larger variation in performance and False Negative Error reduction than would be obtained using Simple Random Sampling. Nevertheless, if the ordering is independent of the data (features and labels), then by the indications of Cochran (1977) [11] the systematic sample will approximate an SRS sample in performance.

4.3 Random Oversampling (ROS)

Now that the performance of Random Undersampling techniques have been analyzed, we turn our attention to the second method of dealing with imbalanced datasets as discussed by Wongvorachan et al. (2023): *oversampling* techniques [8]. The simplest such oversampling technique discussed is termed *Random Oversampling* (ROS), and involves randomly selecting and duplicating elements (*with replacement*) from the minority class until the oversampled minority class has the same size as the majority class [8]. As such, Random Oversampling techniques use resampling with replacement to generate a *larger* balanced dataset than the original dataset/population. Furthermore, based on the indications of Wongvorachan et al. (2023) [8], we assume that the original minority class is included in the oversampled dataset before any resampling is performed, so that each element in the minority class will be included in the oversampled dataset at least once; this would help reduce the problem of information loss and overfitting. As such, the Random Oversampling technique only generates an (additional) sample of size $N_0 - N_1$.

Because ROS functions using resampling with replacement, based on the indications of Cochran (1977) [11] we use the *times* count function t_{ij} instead of a simple indicator function since elements in the minority class appear more than once. Since all elements of the majority class are sampled exactly once in Random Oversampling, the times count function for a feature vector x_{0j} in the majority class is $t_{0j} = 1$, and there is no randomness involved. On the other hand, there *is* randomness in resampling elements from the minority class, so $t_{1j} \in \{0, 1, 2, 3, ..., N_0 - N_1\}$ for a feature vector x_{1j} in the minority class, such that $\sum_{j=1}^{N_1} t_{1j} = N_0 - N_1$. Hence, the expected value of this times count function for the minority class is $E[t_{1j}] = \frac{N_0 - N_1}{N_1}$.

As such, we will investigate two different types of resampling methods for use in Random Oversampling methodologies: Simple Random Sampling with Replacement and Systematic Resampling.

4.3.1 SRS with Replacement

First, we investigate the case in which the resampling technique is Simple Random Sampling with Replacement, to generate a sample of size $N_0 - N_1$ from the overall minority class of size N_1. As such, we determine that the sampling probability depends on the class from which it is selected (corresponding to $y = 0$ and $y = 1$, respectively), but that the sampling within each class is independent of the feature vector x; by extension, we determine that the probability of sampling an element from the imbalanced dataset using Random Oversampling is *independent of* x given y.

Before studying the ROS classifier conditional probabilities, we first define the classifier conditional probability for the original (imbalanced) dataset, which takes the form

$$P(y = 1|x = x') = \frac{\sum_{j=1}^{N_1} I_{1xj}}{\sum_{j=1}^{N_1} I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} \,. \tag{7}$$

The definitions and proof of Equation 7 are given in Appendix D.

Meanwhile, the classifier conditional probability on the ROS Oversampled dataset takes the form

$$P_{ROS}(y = 1|x = x') = \frac{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj}}{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} \,. \tag{8}$$

The proof of Equation 8 is given in Appendix D at the end of the article.

Recalling that $t_{1j} \geq 0$ and comparing the ROS classifier conditional probability in Equation 8 with the imbalanced classifier conditional probability in Equation 7, it can be seen that

$$P_{ROS}(y = 1|x = x') \geq P(y = 1|x = x') \tag{9}$$

In fact, since ROS resamples multiple terms from the minority class (for which $t_{1j} > 0$), Equation 9 yields $P_{ROS}(y = 1|x = x') > P(y = 1|x = x')$ if the feature vector x' is resampled from the minority class. Hence, Random Oversampling reduces the problem of Type II (False Negative) error in Transfer Learning classification.

4.3.2 Systematic Resampling

Lastly, we propose and examine a possible alternative method of conducting Random Oversampling: *Systematic Resampling*. The methodology for this is based on the systematic sampling method outlined by Cochran (1977) [11], and involves taking multiple 1-in-k systematic samples from the minority class and duplicating the elements in that sample. We would then repeat this systematic sampling from the minority class while selecting the beginning point of the systematic sample randomly between 1 and k *with replacement*. For convenience, we assume that (1) both majority and minority class population sizes N_0 and N_1 are integer multiples of k; and (2) that the sampling interval k is held constant across systematic samples in the Systematic Resampling method. It should be noted that our proposed *systematic resampling* method is distinct from that outlined and discussed by Hol et al. (2006) [22].

Therefore, based on the indications of Cochran (1997) [11], the probability of sampling any particular systematic sample from the minority class is $\frac{1}{k}$, and we take a total of $k * \frac{N_0 - N_1}{N_1}$ systematic samples to obtain an Oversampled dataset. Therefore, Systematic Resampling reduces the computational cost of sampling by requiring a smaller number of randomly-sampled values compared to Simple Random Sampling.

As such, by the indications of Cochran (1977) [11], we separate the minority class into k nonoverlapping clusters separated based on their order, such that the j^{th} cluster C_j contains the feature vectors $x_{1(j+qk)}, \forall q \in \{0, 1, 2, ..., \frac{N_1}{k}\}$ where $j \in \{1, 2, ..., k\}$. As such, since we are now conducting random sampling *with replacement* on these k clusters, we define a new *times count* function t_{1j} denoting the number of times that the j^{th} cluster C_j is resampled in the ROS Systematic Resampler, with $t_{1j} \geq 0$ and $\sum_{j=1}^{k} t_{1j} = k * \frac{N_0 - N_1}{N_1}$.

As such, taking a minority-class indicator I_{1jl} denoting whether feature vector x_{1jl} fulfils the condition $x_{1jl} = x'$ (and similar for majority class), the Systematic Resampling classifier conditional probability is

$$P_{ROS}(y = 1 | x = x') = \frac{\sum_{j=1}^{k} \left((t_{1j} + 1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl} \right)}{\sum_{j=1}^{k} \left((t_{1j} + 1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl} \right) + \sum_{j=1}^{N_0} I_{0j}} \tag{10}$$

Meanwhile, the imbalanced classifier conditional probability takes the form in Equation 7. Therefore, since $t_{1j} \geq 0$, we determine that

$$P_{ROS}(y = 1|x = x') \geq P(y = 1|x = x')\,. \tag{11}$$

The proofs of Equations 10 and 11 are given in Appendix E.

In fact, since ROS resamples multiple clusters from the minority class (for which $t_{1j} > 0$), Equation 11 becomes $P_{ROS}(y = 1|x = x') > P(y = 1|x = x')$ if the j^{th} cluster is resampled from the minority class and contains at least one feature vector that fulfils the condition $x_{1jl} = x'$. This confirms that Systematic Resampling as a method of Random Oversampling will reduce the problem of Type II (False Negative) error in Transfer Learning classification.

However, depending on the ordering of the dataset, Systematic Resampling may inadvertently under- or over-represent a certain feature vector in the oversampled minority class and hence yield a larger variation in performance and False Negative Error reduction than would be obtained using SRS Oversampling; hence, care must be taken if attempting to conduct Systematic Resampling as a Random Oversampling methodology.

5 Applications in Diabetes Prediction

Now that the various Random Undersampling and Oversampling methods have been shown to reduce the problem of Type II (False Negative) error in imbalanced datasets from a theoretical standpoint, we now examine the difference in Transfer learning performance for predicting diabetes risk in low-education demographics from the UCI dataset [5]. In the dataset, the education level is recorded from 1-6 in increasing order of education level, in which 1 denotes no education at all; 2 denotes only elementary education; 3 denotes High School dropout; 4 denotes High School graduate; 5 denotes college dropout; and 6 denotes college graduate or higher [5]. The total number of individuals in each education level in the UCI dataset [5] is summarized in the Table below.

Education	Count
6	107325
5	69910
4	62750
3	9478
2	4043
1	174

As can be seen, there are only 4,217 individuals with an education level of 2 or less (elementary school education at most) out of a total dataset of over 250,000 elements. Therefore, we separate the education levels of 1 and 2 from the rest, and designate them as the *target data*, while the elements in the remaining education levels 3-6 will be designated the *source data*. Plotting the counts of the two classes (diabetes negative and positive) for both the source and target data yields Figure 1. As such, the process of Transfer learning will involve training a binary classification model on the source domain (at least some High School education) to predict diabetes risk in this demographic, then use this pre-trained model to predict diabetes risk in the low-education (at most elementary education) demographic. As can be seen, the target task matches the source task (both being binary classification of diabetes risk), but the target and source domains are different (since they differ in terms of education level); as such, this Transfer Learning design is consistent with the Transductive TL process [4].

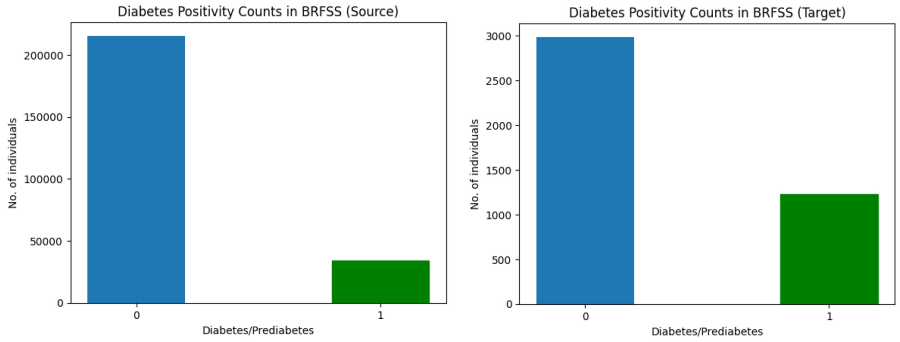

Figure 1: The data imbalance is clearly visible in both source data (left) and target data (right). Interestingly, the low-education group appears to have a smaller imbalance than the higher-education group.

Since Transductive Transfer Learning commonly involves labelled source data but *unlabeled* target data [10], we also treat the target data (for elementary education or less) as being unlabelled (i.e. it is unknown whether they have diabetes). As such, the target data with size of 4,217 elements is not predictable using traditional Machine Learning classification. Instead, we will be using Transductive Transfer Learning to predict diabetes risk in this data using a pre-trained model on the source data; furthermore, to compare the accuracy of prediction, we will obtain a Confusion matrix for the target data to determine the positive and negative error rates of the Transfer Learning classifier.

Since the aim is to obtain a conservative classifier to predict the risk of diabetes or prediabetes from various risk factors, we are primarily interested in reducing the False Negative rate of the classifier. For this reason, we will compare four different Random Undersampling techniques (SRS, SRS with Replacement, single Systematic, and multiple Systematic) and two Random Oversampling techniques (SRS with Replacement, and Systematic Resampling). In particular, we will compile and compare the effectiveness of these sampling techniques in Transductive TL to predict low-education diabetes risk, both in terms of the computational cost (in terms of time) and the predictive error (in terms of the False Negative Rate). For the purposes of training the binary classification model on the source data, we perform a random 80-20 training-testing split of the source data.

To compare the performance of the undersampled/oversampled datasets, we first establish the performance of the imbalanced dataset using Transductive Transfer Learning. Running the TL algorithm yields the Confusion matrices in Figure 2 below. The left-hand Confusion Matrix in Figure 2 denotes the correct/incorrect classification predictions from the source task, obtained in the process of training the model on the source data; as can be seen, the source-trained model has an overall predictive accuracy of approximately 0.866, or 86.6%, but the True Positive Rate is only $TPR_S = 0.1717$, or 17.17%. It can be clearly seen that the imbalanced nature of the training dataset causes a heavy predictive weighting toward the majority group (diabetes negative), making its performance very poor in predicting diabetes risk.

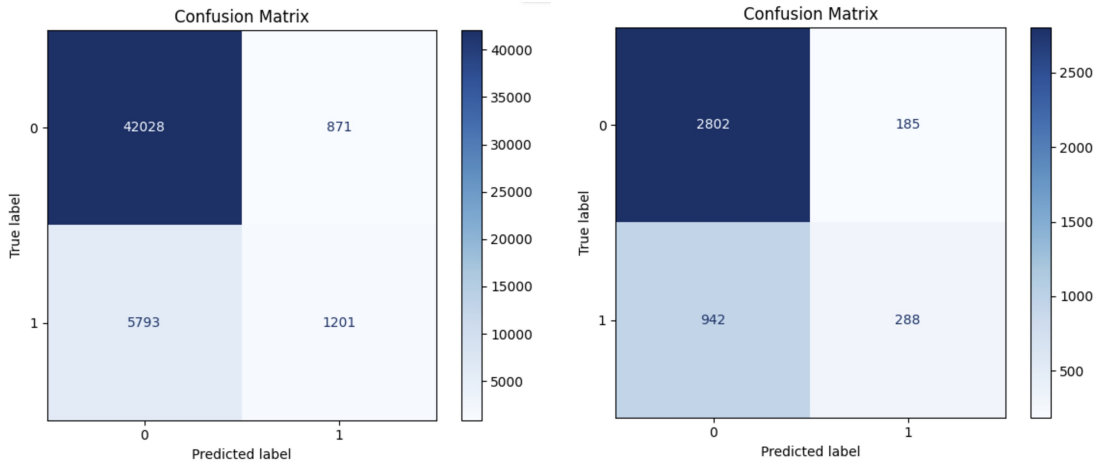

Figure 2: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.1717$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.2341$).

The same problem can be clearly seen in the right-hand confusion matrix of Figure 2, which denotes the predictive confusion matrix of the target data. Namely, although the overall predictive accuracy is approximately 0.7327, or 73.27%, the True Positive Rate is only $TPR_T = 0.2341$, or 23.41%. As such, the imbalanced dataset causes an extremely high False Negative Error of Transductive TL in predicting diabetes risk, making it effectively useless for this purpose.

Given these clear problems in Transfer Learning on the imbalanced dataset, the aim of Random Under/Oversampling is to reduce this False Negative Error in Transductive TL so as to help forewarn which members of the low-education demographic are at risk for the disease. We begin by investigating different methods of Random Undersampling, to compare the performance therein.

5.1 Random Undersampling (RUS)

5.1.1 Simple Random Sampling

The first sampling method we examine is Simple Random Sampling *without* replacement in performing Random Undersampling of the majority class. For this purpose, from the source data we collected a random sample of the same size as the minority class from the majority class without replacement. We then compiled the undersampled majority

class and minority class together to obtain a new undersampled source dataset, and then performed Transductive TL by first training on this new balanced dataset and then predicting the diabetes risk of the low-education demographic. It should be noted that we did not perform any undersampling of the Target dataset, since we treat it as though the label (and hence class membership) is unknown in the target data.

Running the Transductive TL algorithm on this SRS RUS dataset, we get the confusion matrices in Figure 3. As can be seen, the left-hand confusion matrix has a correct predictive ratio of 0.7446, and a True Positive Rate of $TPR_S = 0.8421$ or 84.21%. Meanwhile, the right-hand confusion matrix has a True Positive Rate of $TPR_T = 0.9455$ or 94.55%, which is much larger than for the results from the original imbalanced source dataset in Figure 2. This improvement in diabetes-positive performance is not without its drawback, however; namely, the right-hand confusion matrix shows that the False Positive Rate is now $FPR_T = 0.6733$, indicating that the Transductive TL is now heavily weighted towards predicting diabetes positivity.

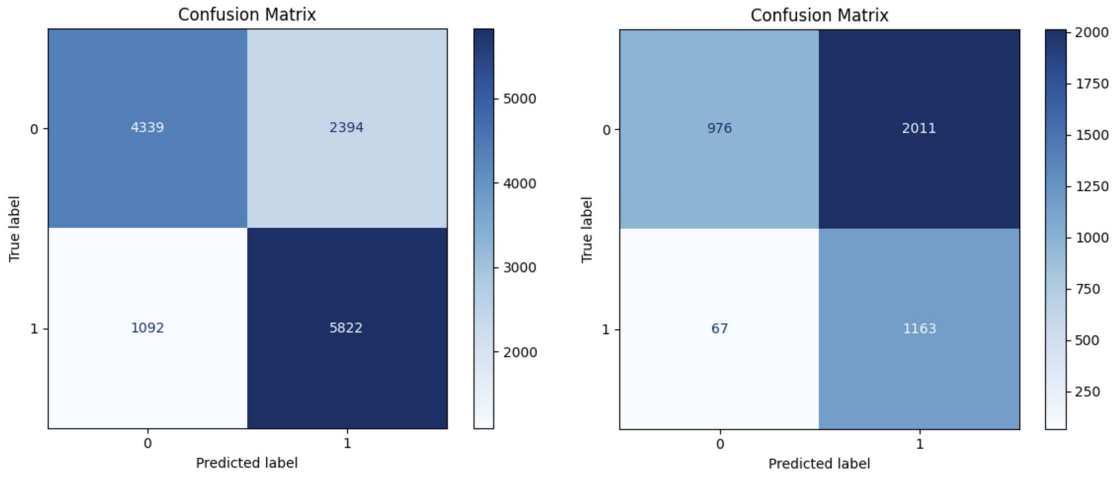

Figure 3: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.8421$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.9455$).

5.1.2 Simple Random Sampling with Replacement

The second sampling method we examine is Simple Random Sampling *with* replacement in performing Random Undersampling of the majority class. For this purpose, from the source data we collected a random sample of the same size as the minority class

from the majority class with replacement. We then compiled the undersampled majority class and minority class together to obtain a new undersampled source dataset, and then performed Transductive TL by first training on this new balanced dataset and then predicting the diabetes risk of the low-education demographic.

Running the Transductive TL algorithm on this SRSwR RUS dataset, we get the confusion matrices in Figure 4. The left-hand confusion matrix has a correct predictive ratio of 0.7498, and a True Positive Rate of $TPR_S = 0.8448$. Meanwhile, the right-hand confusion matrix has a True Positive Rate of $TPR_T = 0.9366$ or 93.66%, which is much larger than for the results from the original imbalanced source dataset in Figure 2. However, the right-hand confusion matrix yields a False Positive Rate of $FPR_T = 0.6562$, indicating that the Transductive TL is weighted towards predicting diabetes positivity.

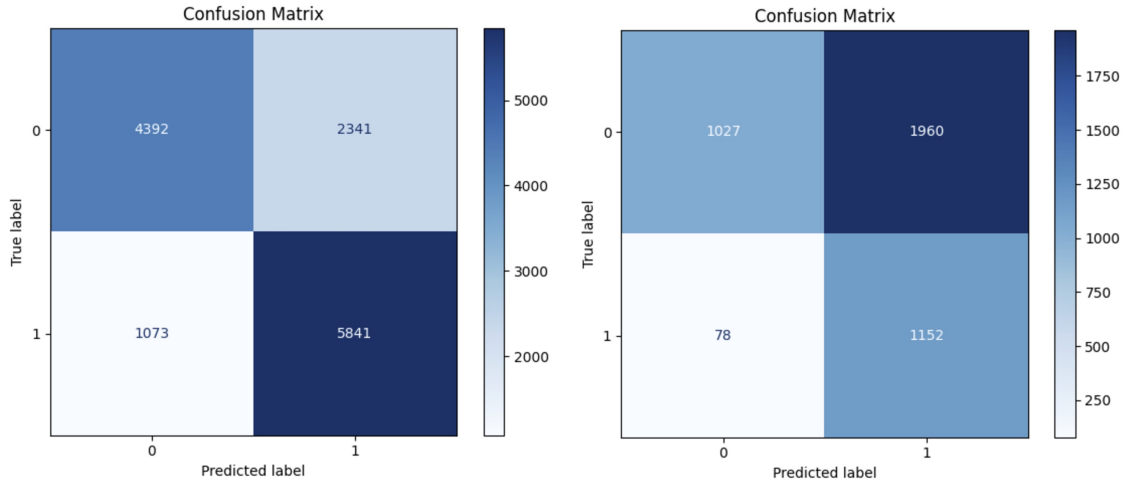

Figure 4: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.8448$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.9366$).

5.1.3 Single Systematic Sample

The third Random Undersampling method we examine is Single Systematic Sampling of the majority class. For this purpose, we take a systematic sampling interval $k = \frac{N_0}{N_1}$, where N_0 is the size of the majority class from the source data and N_1 is the size of the minority class in the source data. For the UCI dataset, this yielded a k-value of $k \approx 6.31$, so we randomly selected a starting point between 1 and 6, then took every subsequent

k^{th} row (calculated as the nearest integer to the multiple of k) to yield an undersampled majority class of the same size as the minority class.

Running the Transductive TL algorithm on this Systematic RUS dataset, we get the confusion matrices in Figure 5. The left-hand confusion matrix has a correct predictive ratio of 0.7466, and a True Positive Rate of 0.8034 or 80.34%. Meanwhile, the right-hand confusion matrix has a True Positive Rate of $TPR_T = 0.9317$ or 93.17%, which is much larger than for the results from the original imbalanced source dataset in Figure 2. However, the right-hand confusion matrix shows that the False Positive Rate is now $FPR_T = 0.6421$, indicating that the Transductive TL is weighted towards predicting diabetes positivity.

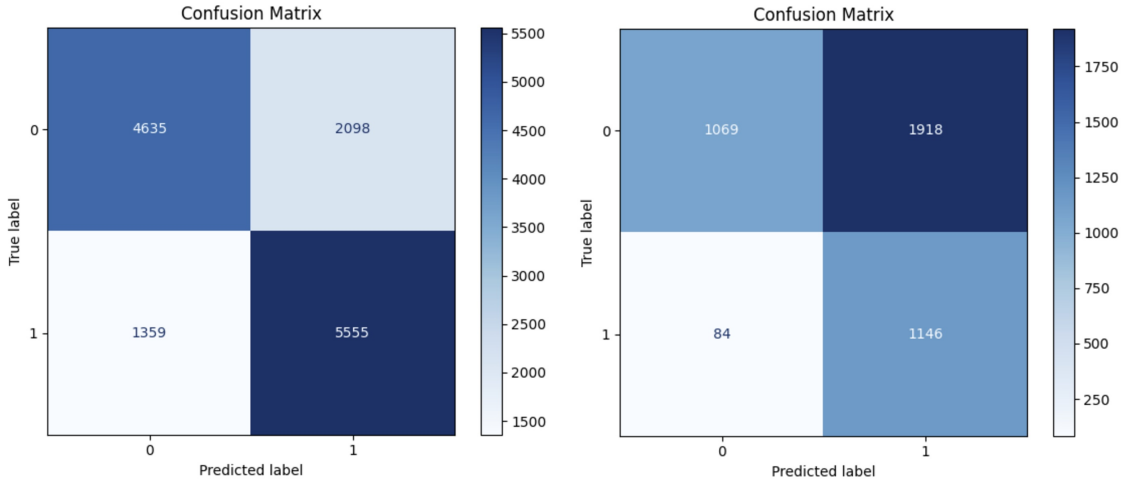

Figure 5: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.8034$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.9317$).

5.1.4 Multiple Systematic Sample

The fourth and final Random Undersampling method we examine is Multiple Systematic Sampling of the majority class, in which we constructed the undersampled dataset using several systematic samples from the majority class. For this purpose, we take a systematic sampling interval $k = m * \frac{N_0}{N_1}$, where m is the number of systematic samples conducted, N_0 is the size of the majority class from the source data and N_1 is the size of the minority class in the source data. For the UCI dataset, we took $m = 10$, which yielded a k-value of $k \approx 63.1$, so we randomly selected 10 random starting points between

1 and 63 with replacement, then took every subsequent k^{th} row (calculated as the nearest integer of multiples of k) for each starting point; doing so and compiling the $m = 10$ systematic samples yielded an undersampled majority class of the same size as the minority class.

Running the Transductive TL algorithm on this Multiple Systematic RUS dataset, we get the confusion matrices in Figure 6. As can be seen from the left-hand confusion matrix has a correct predictive ratio of 0.7496, and a True Positive Rate of 0.74090 or 74.09%. Meanwhile, the right-hand confusion matrix has a True Positive Rate of $TPR_T = 0.9073$ or 90.73%, which is much larger than for the results from the original imbalanced source dataset in Figure 2. On the other hand, the False Positive Rate is now $FPR_T = 0.5815$.

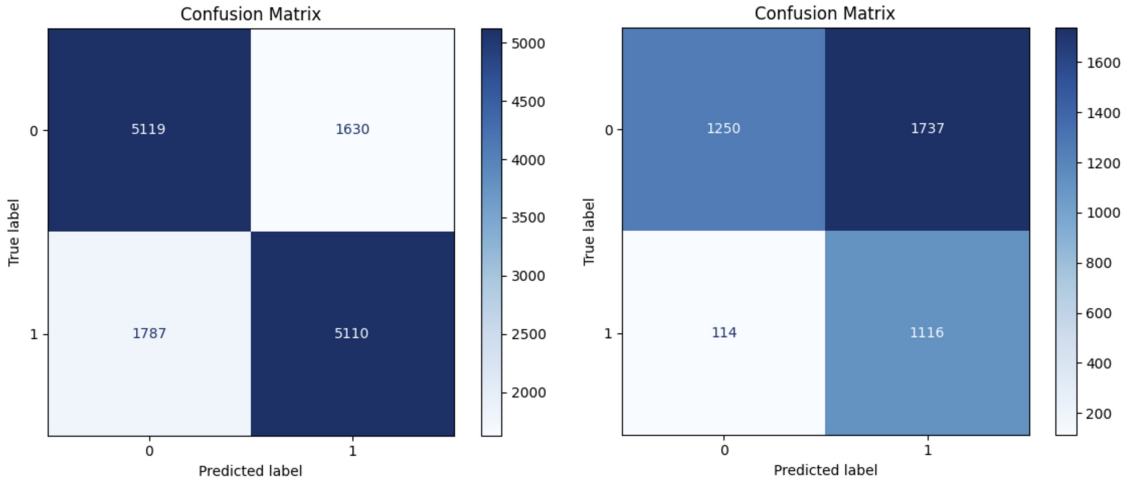

Figure 6: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.7409$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.9073$).

5.2 Random Oversampling (ROS)

Now, we turn our attention to examining the performance of Random Oversampling methods. In particular, we will examine the performance of Simple Random Sampling with Replacement, and Systematic Resampling, as oversampling techniques.

5.2.1 Simple Random Sampling with Replacement

In this case, we perform Random Oversampling (ROS) using Simple Random Sampling with Replacement. To do so, we wish to resample the minority class of size N_1 to obtain additional data of size $N_0 - N_1$; compiling the original minority class with this resampled data therefore yields an SRS Oversampled dataset that is balanced between majority and oversampled minority classes.

Performing this Random Oversampling method, splitting the source data into training and testing sets, and using it to compute low-education diabetes risk using the Transductive TL algorithm, we obtain the Confusion Matrices in Figure 7. As can be observed from the left-hand confusion matrix in Figure 7, the True Positive Rate on the source data is $TPR_S = 0.7842$, while from the right-hand confusion matrix the True Positive rate for the target data using Transductive TL is $TPR_T = 0.92195$.

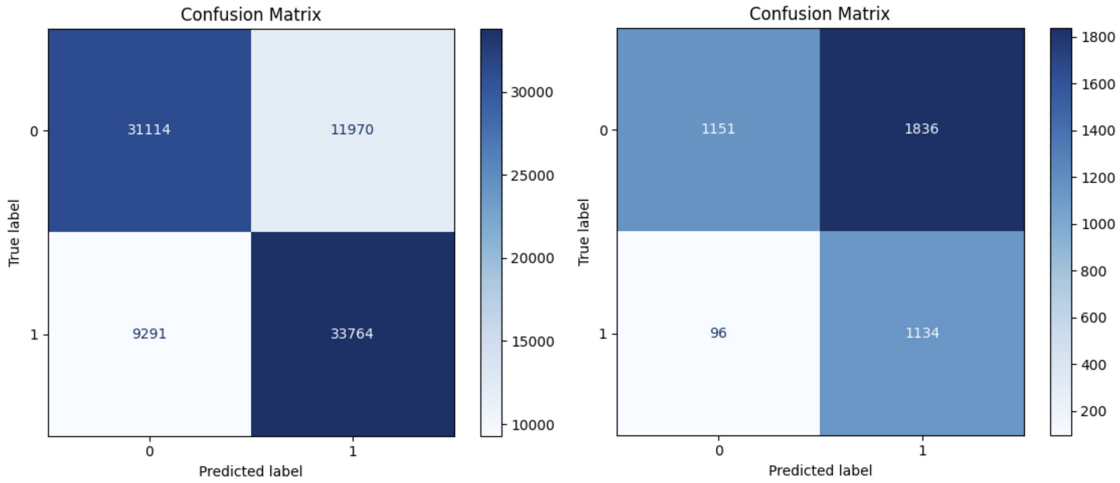

Figure 7: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.7842$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.92195$).

5.2.2 Systematic Resampling

Finally, we turn our attention to our proposed Random Oversampling method, termed *Systematic Resampling*. Namely, taking a systematic sampling interval of k, we obtain a total of $k * \frac{N_0 - N_1}{N_1}$ number of 1-in-k systematic samples (each with random starting point sampled *with replacement*) from the minority class of size N_1. We then compiled these

systematic samples, along with the original minority class, to obtain an oversampled minority class that is (approximately) the same size as the majority class. In our case, we elected to take a systematic sampling interval of $k = 48$.

Running the Transductive TL algorithm on this Systematic Resampling ROS dataset then yielded the confusion matrices in Figure 8. As can be observed from the left-hand confusion matrix in Figure 8, the True Positive Rate on the source data is $TPR_S = 0.8027$, while from the right-hand confusion matrix the True Positive rate for the target data using Transductive TL is $TPR_T = 0.9236$.

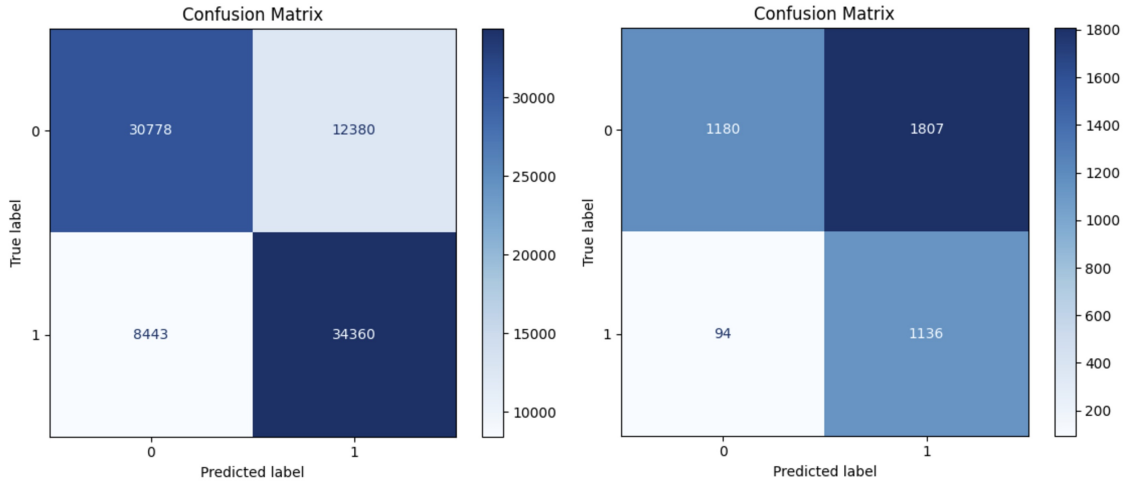

Figure 8: The left-hand Confusion Matrix denotes the predictive results of the source-trained model ($TPR_S = 0.8027$); the right-hand Confusion Matrix denotes the predictive outcome of the Transductive TL algorithm for low-education Diabetes risk ($TPR_T = 0.9236$).

Overall, running 40-50 iterations of the Transductive TL algorithm for each of the six under/oversampling techniques, we obtain the following results:

- **SRS Random Undersampling:** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.07498$ and a 95% Confidence Interval of [0.07082, 0.07915]. Meanwhile, the mean source-training time is 42.972 seconds with a 95% Confidence Interval of [41.11062, 44.83341].

- **SRSwR Random Undersampling:** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.07323$ and a 95% Confidence Interval of [0.06925, 0.07721]. Meanwhile, the mean source-training time is 45.01087 seconds with a 95% Confidence Interval of [43.89413, 46.12761].

- **Systematic Random Undersampling:** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.06928$ and a 95% Confidence Interval of [0.06604, 0.07252]. Meanwhile, the mean source-training time is 50.56252 seconds with a 95% Confidence Interval of [49.38903, 51.73601].

- **Multiple Systematic Random Undersampling:** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.07328$ and a 95% Confidence Interval of [0.06878, 0.07779]. Meanwhile, the mean source-training time is 46.96827 seconds with a 95% Confidence Interval of [45.84260, 48.09394].

- **SRS Random Oversampling:** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.07493$ and a 95% Confidence Interval of [0.07133, 0.07853]. Meanwhile, the mean source-training time is 231.34685 seconds with a 95% Confidence Interval of [222.30849, 240.38521].

- **Systematic Resampling (Oversampling):** The mean False Negative Rate (FNR) is $\hat{FNR} = 0.07730$ and a 95% Confidence Interval of [0.07361, 0.08098]. Meanwhile, the mean source-training time is 216.44402 seconds with a 95% Confidence Interval of [209.20947, 223.67858].

As can be seen, there is no significant difference (with some exceptions) between the mean False Negative Rates for the six different under/oversampling techniques, so we conclude that all the techniques yield (approximately) equivalent performance in reducing Type II error. However, it can be clearly seen that the SRS RUS technique had a significantly lower mean training time than the Systematic RUS and multiple systematic RUS, as well as both Oversampling methods (as would be expected). Overall, we conclude that Simple Random Sampling as an Undersampling technique performs the most efficiently (in terms of computational cost) while yielding (almost) equivalent performance. This indicates that SRS Undersampling is the optimal sampling technique for Transductive TL prediction of diabetes risk.

6 Predicting Heart Disease using Fine-tuning

Lastly, we turn our attention to another application of Transfer Learning: that of *fine-tuning* a pre-trained model [4] on the target data. Unlike the previous section, we now elect to take two different tasks for the source and target task: namely, the source task is to predict diabetes risk from the high-education population (now comprising of education levels 4-6 in the UCI dataset [5]) while the target task is to predict the incidence of heart disease or heart attack from the low-education demographics (education level of 1-3 in [5]).

For this purpose, we first split the data into the high-education (4-6) and low-education (1-3) groups, respectively, and designate the much larger high-education group to be the source data, while the low-education group corresponds to the target data. We also drop the "HeartDiseaseorAttack" column from the source data, and keep Diabetes as the source label. Meanwhile, we drop the diabetes label from the target data and use "HeartDiseaseorAttack" as the target label. Unlike the Transductive TL algorithm used in Section 5 previously, here we assume that we know the labels for the target data in order to perform fine-tuning.

Once this is done, we split both the source and target data into training-testing sets with 80-20 splits each, and create a new multi-layer DNN for source training. We then train this DNN on the source training and testing data. We then use this pre-trained model to fine-tune to the target training data by only setting the last two training layers of the model to be trainable, while keeping all other layers fixed from the pre-training. As can be seen in Figure 9, we observe that the target training set is highly imbalanced, which may warrant the use of under/oversampling to yield better fine-tuning predictive performance of heart disease.

Training the model on the original (imbalanced) source data and then fine-tuning it on the imbalanced target training data, we obtain the Confusion Matrix in Figure 10. Most importantly, we observe that the fine-tuned model does not predict *any* points to be positive with regards to heart disease, thereby showing a critical failure in heart disease prediction using the source and target imbalanced data.

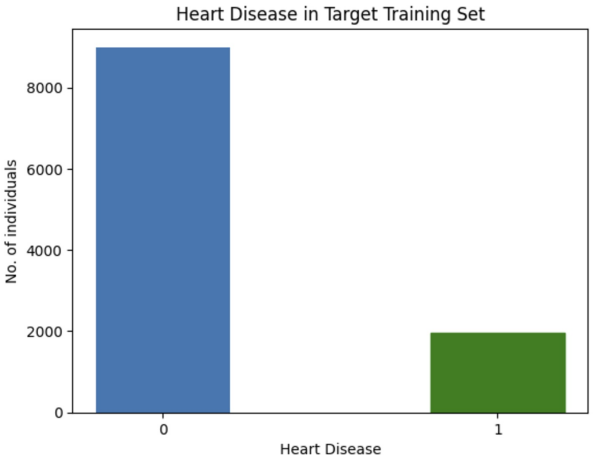

Figure 9: The target training set is highly imbalanced, which may cause problems in fine-tuning.

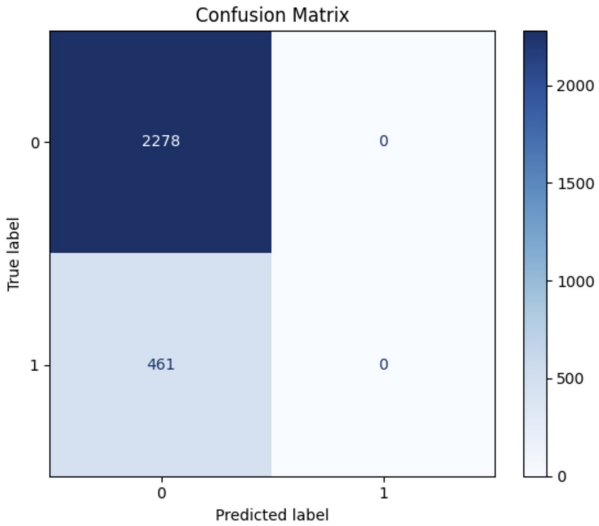

Figure 10: As can be seen, there are *no* points predicted as positive on heart disease on this imbalanced dataset using fine-tuning.

Once this was done, we then performed pre-training on the balanced source data, using the six Random Undersampling and Random Oversampling techniques discussed in Section 5. However, although these showed improvements in performance before fine-tuning, the incorporation of the imbalanced target training data during the fine-tuning stage completely undid any performance gains, such that the Confusion Matrices even with pre-training on balanced data took forms very similar to that in Figure 10.

As such, we conclude that the Transfer Learning fine-tuning algorithm also requires some form of sampling technique on its target training data in order to balance it and reduce the False Negative error of the model. As such, for this purpose we include the same under/oversampling techinques for both the source data and the target training data, although we do not attempt any balancing for the target testing data. Since the under/oversampling techniques only apply to the target *training* (not testing) data, there is no problem of training/testing overlap in data. Performing the fine-tuning on these balanced datasets (using SRS RUS, SRSwR RUS, Systematic RUS, multiple systematic RUS, SRS ROS, and Systematic Resampling, respectively) yields the confusion matrices in Figure 11.

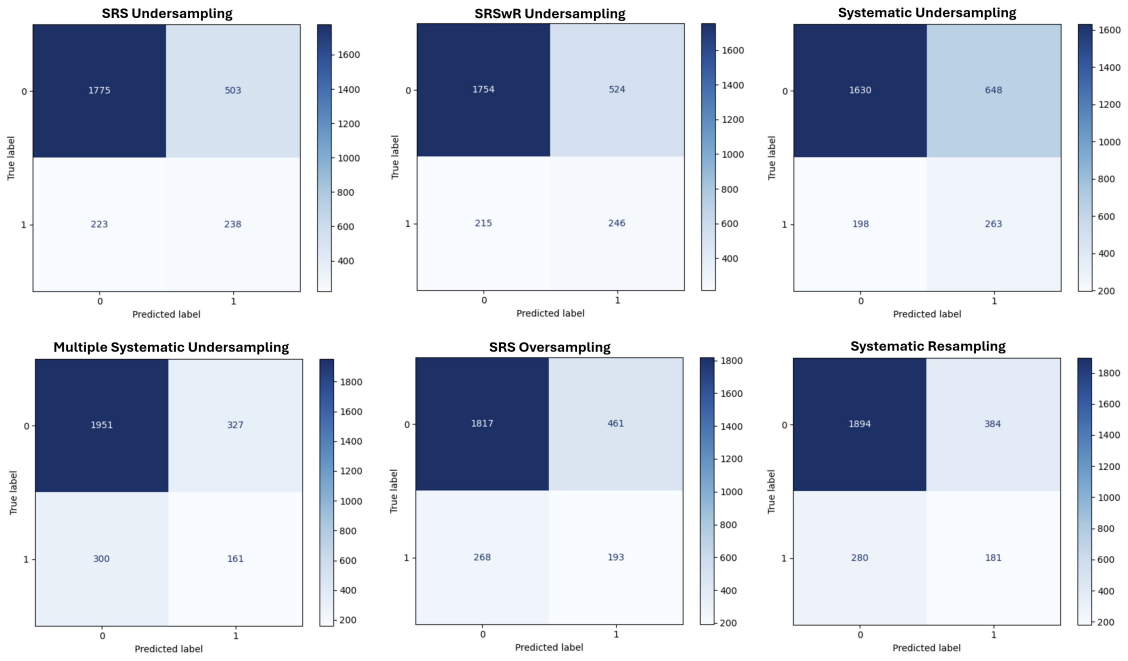

Figure 11: Confusion matrices obtained using fine-tuning on balanced target training dataset for each under/oversampling method.

To obtain an estimate of the performance of each sampling method in fine-tuning, we ran the fine-tuning algorithms 50 times each to compile the False Positive and False Negative error rates. Doing so, we obtained the following performance results:

- **SRS Random Undersampling:** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.2118$ and a 95% Confidence Interval of [0.20065, 0.22295]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.45957$ and a 95% Confidence Interval of [0.44199, 0.47715].

- **SRSwR Random Undersampling:** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.17845$ and a 95% Confidence Interval of [0.16055, 0.19635]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.58373$ and a 95% Confidence Interval of [0.55137, 0.61609].

- **Systematic Random Undersampling:** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.15563$ and a 95% Confidence Interval of [0.14419, 0.16706]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.66187$ and a 95% Confidence Interval of [0.64180, 0.68193].

- **Multiple Systematic Random Undersampling:** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.18863$ and a 95% Confidence Interval of [0.17402, 0.20324]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.49970$ and a 95% Confidence Interval of [0.47731, 0.52208].

- **SRS Random Oversampling:** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.19921$ and a 95% Confidence Interval of [0.19033, 0.20809]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.50256$ and a 95% Confidence Interval of [0.48781, 0.51731].

- **Systematic Resampling (Oversampling):** Performing 50 iterations, we obtain a mean False Positive Rate (FPR) of $\hat{FPR} = 0.16856$ and a 95% Confidence Interval of [0.15513, 0.18199]. Meanwhile, we obtain a mean False Negative Rate (FNR) of $\hat{FNR} = 0.60950$ and a 95% Confidence Interval of [0.58515, 0.63386].

As can be observed, the False Negative mean rate for SRS Undersampling (with mean of 0.45957) is significantly lower than the False Negative rates for all the other under/oversampling techniques. This indicates that SRS Undersampling is the best method to employ in reducing False Negative Error in fine-tuning a Transfer Learning model. In order of mean False Negative Rate (from best to worst), the sampling techniques are: (1) SRS Undersampling; (2) Multiple Systematic Undersampling; (3) SRS Oversampling; (4) SRSwR Undersampling, (5) Systematic Resampling/Oversampling; and (6) Systematic Random Undersampling. It would appear that, despite their ease of application [11], systematic sampling techniques perform worse than Simple Random Sampling techniques in helping reduce False Negative Error on imbalanced target training data.

7 Conclusion

Overall, in this project we discussed and analyzed various sampling techniques and their effect on reducing Type II (False Negative) error in DNN-based Transfer Learning. Since we elected to apply the binary classification Transfer Learning algorithm to the "CDC Diabetes Health Indicators" dataset in the UCI Machine Learning repository [5], the sampling techniques proved necessary to predicting diabetes risk given the severe imbalance of the dataset.

As such, in Section 4, we analyzed several different sampling/resampling techniques for the application of Random Undersampling (RUS) and Random Oversampling (ROS) as outlined by Wongvorachan et al. (2023) [8]. For Random Undersampling, we examined three different sampling methods to balance the dataset: Simple Random Sampling, Simple Random Sampling with Replacement, and Systematic Sampling, and showed from a theoretical standpoint (using the Bayes classifier) why undersampling the dataset helped to decrease the problem of Type II (False Negative) error in binary classification. Similarly, for Random Oversampling, we examined two different sampling methods: Simple Random Sampling with Replacement and Systematic Resampling, and showed from a theoretical standpoint why these resampling techniques also help reduce Type II error in

binary classification.

Following this, in Section 5, we then applied these sampling techniques to the UCI dataset to observe whether, and to what degree, predictive performance improved in Transductive Transfer Learning for binary classification of diabetes risk in low-education demographics. Running the Transfer Learning algorithm on the original (imbalanced) dataset, we observed a mean source-training time of 167.2 seconds and a mean True Positive Rate of $TPR_T = 0.216$; this performance failure is consistent with our expectation from Liu et al. (2023) [7].

We then applied four different *Random Undersampling* techniques: Simple Random Sampling without replacement; Simple Random Sampling with replacement; Single Systematic Sample; and Multiple Systematic Sample. We further applied two different *Random Oversampling* techniques: Simple Random Sampling with Replacement; and Systematic Resampling. Doing so, we observed that there is no significant difference in TPR/FNR predictive accuracy between these RUS and ROS methods, but that SRS Undersampling had a significantly lower training time than the Systematic Undersampling and Oversampling techniques. This indicates that, for the purposes of computational efficiency it is preferable to conduct SRS Undersampling for Transductive TL, as this will yield equivalent predictive performance while reducing the necessary training time.

Finally, in Section 6, we constructed a multi-layer DNN and used fine-tuning to predict Heart disease risk in low-education demographics using diabetes in high-education demographics. We observed that, despite training the DNN on a balanced dataset (using the six under/oversampling techniques discussed previously), the imbalanced target training set caused extremely poor predictive performance for heart disease risk. As such, we applied and investigated the fine-tuning performance for each of the six under/oversampling techniques on both the source data and the target training data. In so doing, we observed that the SRS Undersampling technique yielded a significantly lower mean False Negative Error rate compared to the other under/oversampling techniques. Therefore, we conclude that Simple Random Sampling as an Undersampling technique is preferable for reducing False Negative error in fine-tuning on imbalanced data. Overall,

the results of both Sections 5 and 6 indicate that SRS Undersampling is the optimal sampling technique to use in improving Transfer Learning performance on imbalanced datasets.

Overall, while this study serves as an interesting foray into the topic of sampling in improving Transfer Learning effectiveness, there are still some limitations and possible areas of further development of this project. Firstly, it is possible that the lack of significant difference in performance between the under/oversampling methods in Section 5 is caused by the (relatively) small size of the undersampled dataset; as such, it would be instructive to apply this Transfer Learning algorithm to a much larger dataset (consisting of millions of elements/datapoints), to observe whether, and how, training time differs on larger datasets. Secondly, given more time it would be instructive to obtain a much larger set of iterations for the Transductive TL in Section 5, to determine whether a superior sampling technique (in terms of False Negative Rate) can be identified. Thirdly, while in Section 6 we applied the same under/oversampling technique to both the source data and target training set, it would be of interest to investigate how different combinations of sampling techniques would influence predictive performance in fine-tuning a Deep Neural Network. Overall, the primary contribution of this study was to outline and investigate the performance of different sampling techniques in Transfer Learning on imbalanced data, in which we observed that Simple Random Undersampling still had consistently better performance than other possible sampling and resampling techniques.

The code used in this project, along with the resulting performance data and supplementary codes, can be found at: `https://github.com/AndreiAf02/STAT561_Project/tree/main`

References

[1] A. ElRafey and J. Wojtusiak. Recent advances in scaling-down sampling methods in machine learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(6), 2017. https://doi.org/10.1002/wics.1414.

[2] K. Weiss, T.M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *J Big Data*, 3(9), 2016. https://doi.org/10.1186/s40537-016-0043-6.

[3] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. https://doi.org/10.1109/JPROC.2020.3004555.

[4] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim. Transfer learning: a friendly introduction. *J Big Data*, 9(1), 2022. https://doi.org/10.1186/s40537-022-00652-w.

[5] Alex Teboul. CDC Diabetes Health Indicators. Retrieved November 6, 2024, from https://doi.org/10.24432/C53919.

[6] Centres for Disease Control and Prevention. Behavioral Risk Factor Surveillance System. Retrieved November 6, 2024, from https://www.cdc.gov/brfss/index.html.

[7] Yang Liu, Guoping Yang, Shaojie Qiao, Meiqi Liu, Lulu Qu, Nan Han, Tao Wu, Guan Yuan, Tao Wu, and Yuzhong Peng. Imbalanced data classification: Using transfer learning and active sampling. *Engineering Applications of Artificial Intelligence*, 117:105621, 2023. https://doi.org/10.1016/j.engappai.2022.105621.

[8] Tarid Wongvorachan, Surina He, and Okan Bulut. A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining. *Information*, 14(1), 2023. https://doi.org/10.3390/info14010054.

[9] G. Ayana, K. Dese, A.M. Abagaro, Kwangcheol Casey Jeong, Soon-Do Yoon, and Se woon Choe. Multistage transfer learning for medical images. *Artificial Intelligence Review*, 57(232), 2024. https://doi.org/10.1007/s10462-024-10855-7.

[10] IBM. Historical Data. Retrieved October 30, 2024, from https://www.ibm.com/topics/transfer-learning.

[11] William G. Cochran. *Sampling Techniques*. Wiley, 3rd edition, 1977.

[12] Guillaume Chauvet and Audrey-Anne Vallée. Inference for two-stage sampling designs. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(3):797–815, 05 2020. https://doi.org/10.1111/rssb.12368.

[13] C. F. J. Wu, D. Holt, and D. J. Holmes. The Effect of Two-Stage Sampling on the F Statistic. *Journal of the American Statistical Association*, 83(401):150–159, 1988. https://doi.org/10.2307/2288934.

[14] Alex Teboul. Diabetes Health Indicators Dataset. Retrieved November 6, 2024, from https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset.

[15] Andy Peytchev, Lisa R. Carley-Baxter, and Michele C. Black. Multiple sources of nonobservation error in telephone surveys: Coverage and nonresponse. *Sociological Methods & Research*, 40(1):138–168, 2011. https://doi.org/10.1177/0049124110392547.

[16] Centres for Disease Control and Prevention. 2023 Summary Data Quality Report. Retrieved November 6, 2024, from https://www.cdc.gov/brfss/annual_data/2023/pdf/2023-DQR-508.pdf.

[17] Centres for Disease Control and Prevention. Weighting the BRFSS Data. Retrieved November 6, 2024, from https://www.cdc.gov/brfss/annual_data/2023/pdf/2023-Weightning-Description-508.pdf.

[18] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. 21st International Conference on Machine Learning, page 114, New York, NY, USA, 2004. Association for Computing Machinery. https://doi.org/10.1145/1015330.1015425.

[19] J. W. Horn, T. Feng, B. Mørkedal, D. Aune, L. B. Strand, J. Horn, K. J. Mukamal, and I. Janszky. Body mass index measured repeatedly over 42 years as a risk factor for ischemic stroke: The hunt study. *Nutrients*, 15(5):1232, 2023. https://doi.org/10.3390/nu15051232.

[20] Xinyu Wang, Yanan Huang, Yanru Chen, Tingting Yang, Wenli Su, Xiaoli Chen, Fanghong Yan, Lin Han, and Yuxia Ma. The relationship between body mass index and stroke: a systemic review and meta-analysis. *Journal of Neurology*, 269:6279–6289, 2022. https://doi.org/10.1007/s00415-022-11318-1.

[21] K. S. Alwadeai, M. A. Almeshari, A. S. Alghamdi, A. M. Alshehri, S. S. Alsaif, M. O. Al-Heizan, M. S. Alwadei, A. D. Alahmari, S. S. Algarni, T. F. Alotaibi, M. M. Alqahtani, N. Alqahtani, J. S. Alqahtani, A. M. Aldhahir, M. M. Homoud, and S. A. Alhammad. Relationship Between Heart Disease and Obesity Indicators Among Adults: A Secondary Data Analysis. *Cureus*, 15(3):e36738, 2023. https://doi.org/10.7759/cureus.36738.

[22] Jeroen D. Hol, Thomas B. Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82, 2006. https://doi.org/10.1109/NSSPW.2006.4378824.

Appendix A: Bayesian Classifier Conditional Probabilities

We investigate the first three cases outlined by Zadrozny (2004): (1) the sampling I is altogether independent of the values of x and y; (2) the sampling is independent of y *given* x (i.e. $P(I|x,y) = P(I|y)$); (3) the sampling is independent of x *given* y (i.e. $P(I|x,y) = P(I|x)$) [18].

Case 1: Taking the sampling I to be independent of both x and y, we have $P(y|x, I = 1) = P(y|x)$, $P(x|y, I = 1) = P(x|y)$, $P(y|I = 1) = P(y)$, and $P(x|I = 1) = P(x)$, so the classifier conditional probability in the sample is trivially equal to the conditional probability for the overall dataset/population; in mathematical terms,

$$P(y|x, I = 1) = \frac{P(x|y, I = 1)P(y|I = 1)}{P(x|I = 1)} = \frac{P(x|y)P(y)}{P(x)} = P(y|x)\,.$$

Case 2: Taking the sampling to be independent of y given x (such that $P(I = 1|x, y) = P(I|x)$), we have $P(I = 1|x, y) = \frac{P((I=1)\cap x\cap y)}{P(x\cap y)}$ and that $P(y|x, I = 1) = \frac{P((I=1)\cap x\cap y)}{P(x\cap(I=1))}$, so we have $P(y|x, I = 1) = \frac{P(I=1|x,y)*P(x\cap y)}{P(x\cap(I=1))}$.

In that case, knowing that $P(x \cap (I = 1)) = P(I = 1|x) * P(x)$ and that $P(x \cap y) = P(y|x) * P(x)$, we get

$$P(y|x, I = 1) = \frac{P(I = 1|x, y) * P(x \cap y)}{P(x \cap (I = 1))} = \frac{P(I = 1|x) * P(y|x) * P(x)}{P(I = 1|x) * P(x)} = P(y|x)$$

Therefore, we conclude that if the sampling is independent of y given x, the sampling bias does not have an effect on the classifier conditional probability $P(y|x)$ of the sample versus the overall dataset/population. This matches the conclusion drawn by Zadrozny (2004) [18].

Case 3: Taking the sampling to be independent of x given y (such that $P(I = 1|x, y) = P(I|y)$), we observe that $P(x|y, I = 1) = \frac{P(x\cap y\cap(I=1))}{P(y\cap(I=1)}$, $P(y|I = 1) = \frac{P(y\cap(I=1))}{P(I=1)}$, $P(x|I = 1) = \frac{P(x\cap(I=1))}{P(I=1)}$, in which case Equation 2 becomes

$$P(y|x, I = 1) = \frac{\frac{P(x\cap y\cap(I=1))}{P(y\cap(I=1)} * \frac{P(y\cap(I=1))}{P(I=1)}}{\frac{P(x\cap(I=1))}{P(I=1)}} = \frac{P(x \cap y \cap (I = 1))}{P(x \cap (I = 1))}$$

In that case, we observe that $P(x \cap y \cap (I = 1)) = P(I = 1|x, y) * P(x \cap y)$ and that $P(x \cap (I = 1)) = P(I = 1|x) * P(x)$. Recalling that $P(I = 1|x, y) = P(I = 1|y)$, we determine that the classifier conditional probability of the sample takes the form

$$P(y|x, I = 1) = \frac{P(I = 1|y) * P(x \cap y)}{P(I = 1|x) * P(x)} = \frac{P(I = 1|y)}{P(I = 1|x)} * \frac{P(x \cap y)}{P(x)}$$

Recalling the Bayes estimator in Equation 1, we determine that Case 3 yields

$$P(y|x, I = 1) = P(y|x) * \frac{P(I = 1|y)}{P(I = 1|x)}$$

This yields Equation 3 in the article. □

Appendix B: SRS Undersampling

Here, we investigate the effectiveness of performing Simple Random Sampling *without replacement* to select a balanced subset of the overall dataset. N_0 denotes the majority class size, and N_1 denotes minority class size. Hence, we use the sampling indicator I_{0i} to denote whether the feature vector x_{0i} will be sampled from the majority class, with $I_{0i} = 1$ if x_{0i} is sampled and $I_{0i} = 0$ otherwise, where $E(I_{0i}) = \frac{N_1}{N_0}$.

Therefore, using Equation 3, we determine that the conditional probability $P(I = 1|y) = \frac{P((I=1) \cap y)}{P(y)}$. Since such Random Undersampling functions like Stratified Sampling on label strata ($y = 0$ and $y = 1$, respectively), we write

$$P((I = 1) \cap (y = y')) = \sum_{i=0}^{1} W_i P_i((I = 1) \cap (y = y')) = \frac{1}{N} \sum_{i=0}^{1} \left(N_i * \frac{1}{N_i} \sum_{j=1}^{N_i} I_{yij} I_{ij} \right) = \frac{1}{N} \sum_{i=0}^{1} \sum_{j=1}^{N_i} I_{yij} I_{ij} \,,$$

Where I_{yij} is an indicator function denoting whether label y_{ij} fulfils the condition $y_{ij} = y'$ for either $y' = 0$ or $y' = 1$. If $y' = 1$, then $I_{yij} = y_{ij}$, while if $y' = 0$, then $I_{yij} = 1 - y_{ij}$. Since we are primarily interested in reducing Type II (False Negative) error caused by imbalance, we examine the situation in which $y' = 1$ (predicting diabetes positivity using conditional probability classification).

In that case, $I_{yij} = y_{ij}$ and hence

$$P((I = 1) \cap (y = y')) = \frac{1}{N} \sum_{i=0}^{1} \sum_{j=1}^{N_i} I_{yij} I_{ij} = \frac{1}{N} \sum_{j=1}^{N_1} I_{1j} = \frac{N_1}{N} \,.$$

Furthermore, we have

$$P(y = y') = \frac{1}{N} \sum_{i=0}^{1} N_i * \frac{1}{N_i} \sum_{j=1}^{N_i} y_{ij} = \frac{1}{N} \sum_{i=0}^{1} \sum_{j=1}^{N_i} y_{ij} = \frac{N_1}{N} \,.$$

Therefore, taking $y' = 1$, we obtain

$$P(I = 1 | y = y') = \frac{P((I = 1) \cap (y = y'))}{P(y = y')} = 1 \,.$$

Meanwhile, this undersampling technique will select only some of the feature vectors x for which $x = x'$ in the full dataset/population, so we have $P(I = 1 | x = x') \leq 1, \forall x' \in \chi$. In particular, if the feature vector x' appears in the majority class (though not necessarily exclusively), then $P(I = 1 | x = x') < 1$.

Therefore, the classifier conditional probability using SRS undersampling is

$$P(y = 1 | x = x', I = 1) = P(y = 1 | x = x') * \frac{P(I = 1 | y = 1)}{P(I = 1 | x = x')} = \frac{P(y = 1 | x = x')}{P(I = 1 | x = x')} \geq P(y = 1 | x = x') \,.$$

This yields Equation 5. □

Appendix C: SRSwR Undersampling

Now, we examine the Random Undersampling technique using Simple Random Sampling *with replacement*. In this case, we again select a sample of size N_1 from the majority class of size N_0, and (by the indications of Cochran (1977) [11]) replace the sampling indicator function I_i from the previous section with a *times* function t_i that counts the *number* of times that a particular feature vector x_i is selected.

Once again, since the Random Undersampling methodology takes the entire minority class (corresponding to $y = 1$), it therefore samples all feature vectors x_{1i} from the minority class exactly once, so $t_{1i} = 1$. On the other hand, the feature vector x_{0i} from the majority class will be selected a random number of times t_{0i}, with possible values $t_{0i} \in \{0, 1, 2, ..., N_1\}$, such that $\sum_{i=1}^{N_0} t_{0i} = N_1$. As such, we can write each individual term of the *count* function t_{0i} corresponding to the feature vector x_{0i} in the majority class as $t_{0i} = \sum_{j=1}^{N_1} I_{0ij}$, where I_{0ij} is an indicator function denoting whether term x_{0j} from the *undersampled* majority class is term x_{0i} from the full majority class. Therefore, the property becomes

$$\sum_{i=1}^{N_0} t_{0i} = \sum_{i=1}^{N_0} \sum_{j=1}^{N_1} I_{0ij} = N_1 \, .$$

As in Section 4.2.1, we observe that the sampling probabilities are dependent on the label y, but we assume that the SRS with replacement within each class/stratum is independent of the feature vector x, thereby fulfilling the condition $P(t|x, y) = P(t|y)$ (replacing the indicator I with the count function t). Therefore, Equation 3 becomes

$$P(y = y'|x, t \geq 1) = P(y = y'|x) * \frac{P(t \geq 1|y = y')}{P(t \geq 1|x)}$$

In this case, we have $P(t \geq 1|y = y') = 1 - P(t = 0|y = y')$.

Since we are again concerned with Type II error, we study the case where $y' = 1$ (positive minority class), in which case we know that $t_{ij} = 1$ (since all elements of the minority class are sampled exactly once). Therefore, we determine that the conditional probability of *not* sampling an element in the minority class is 0, so $P(t = 0|y = y') = 0 => P(t \geq 1|y = y') = 1$.

Furthermore, we know that the probability of sampling feature vector x at least once given a particular value of x fulfils the condition $P(t \geq 1|x) \leq 1$, and for feature vectors that are also located in the majority class the condition is $P(t \geq 1|x) < 1$.

Therefore, we determine that the sample classifier conditional probability is

$$P(y = y'|x, t \geq 1) = P(y = y'|x) * \frac{P(t \geq 1|y = y')}{P(t \geq 1|x)} = \frac{P(y = y'|x)}{P(t \geq 1|x)} \geq P(y = y'|x)$$

This yields Equation 6. □

Appendix D: SRS Oversampling

To begin, we first define the classifier conditional probability for the original (imbalanced) dataset. Taking all elements x_{1j} in the minority class (corresponding to $y_{1j} = 1$) that fulfil the condition $x_{1j} = x'$ for a given feature vector x', the total number of times that these elements appear in the original (imbalanced) dataset from the minority class are $\sum_{j=1}^{N_1} I_{1xj}$, where I_{1xj} is an indicator function denoting whether $x_{1j} = x'$. Therefore, since the overall dataset is of size $N = N_0 + N_1$, the probability of $y_j = 1$ *and* $x_j = x'$ is

$$P((y = 1) \cap (x = x')) = \frac{1}{N} \sum_{j=1}^{N_1} I_{1xj} \,.$$

Similarly, taking all elements x_{0j} in the majority class (corresponding to $y_{0j} = 0$) that fulfil the condition $x_{0j} = x'$, the total number of times that these elements appear in the original (imbalanced) dataset from the majority class are $\sum_{j=1}^{N_0} I_{0xj}$, where I_{0xj} is an indicator function denoting whether $x_{0j} = x'$. As such, the probability of $y_j = 0$ *and*

$x_j = x'$ is

$$P((y=0) \cap (x=x')) = \frac{1}{N} \sum_{j=1}^{N_0} I_{0xj} \, .$$

Therefore, the total probability of $x = x'$ in the original imbalanced dataset is

$$P(x=x') = P((y=1) \cap (x=x')) + P((y=0) \cap (x=x')) = \frac{1}{N} \left(\sum_{j=1}^{N_1} I_{1xj} + \sum_{j=1}^{N_0} I_{0xj} \right) .$$

Hence, the classifier conditional probability of $y = 1$ given $x = x'$ from the original (imbalanced) dataset is

$$P(y=1|x=x') = \frac{P((y=1) \cap (x=x'))}{P(x=x')} = \\ = \frac{\frac{1}{N} \sum_{j=1}^{N_1} I_{1xj}}{\frac{1}{N} \left(\sum_{j=1}^{N_1} I_{1xj} + \sum_{j=1}^{N_0} I_{0xj} \right)} = \frac{\sum_{j=1}^{N_1} I_{1xj}}{\sum_{j=1}^{N_1} I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} \, .$$

This yields Equation 7. □

Now, we turn our attention to the Bayesian classifier on the Random Oversampled dataset; for this, we assume that the feature vector and corresponding label (x_{1j}, y_{1j}) in the minority class are resampled a total of t_{1j} times, where $t_{1j} \geq 0$ and $\sum_{j=1}^{N_1} t_{1j} = N_0 - N_1$. Since the oversampled dataset also contains the original minority class in its entirety besides the resampled data, element (x_{1j}, y_{1j}) appears a total of $t_{1j} + 1$ times in the oversampled dataset. Taking all elements x_{1j} in the minority class (corresponding to $y_{1j} = 1$) that fulfil the condition $x_{1j} = x'$ for a given feature vector x', the total number of times that these elements appear in the oversampled dataset from the minority class are $\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj}$, where I_{1xj} is an indicator function denoting whether $x_{1j} = x'$. Therefore, the probability of $y_j = 1$ *and* $x_j = x'$ in the overall oversampled dataset (of size $2N_0$) is

$$P_{ROS}((y=1) \cap (x=x')) = \frac{1}{2N_0} \sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} \, .$$

On the other hand, taking all elements x_{0j} in the majority class (corresponding to

$y_{0j} = 0$) that fulfil the condition $x_{0j} = x'$, the total number of times that these elements appear in the majority class are $\sum_{j=1}^{N_0} I_{0xj}$, where I_{0xj} is an indicator function denoting whether $x_{0j} = x'$. Therefore, the probability of $y_j = 0$ *and* $x_j = x'$ in the overall oversampled dataset (of size $2N_0$) is

$$P_{ROS}((y = 0) \cap (x = x')) = \frac{1}{2N_0} \sum_{j=1}^{N_0} I_{0xj} \, .$$

Hence, the *total* probability of $x = x'$ in the overall oversampled dataset (of size $2N_0$) is

$$P_{ROS}(x = x') = P_{ROS}((y = 1) \cap (x = x')) + P_{ROS}((y = 0) \cap (x = x')) =$$
$$= \frac{1}{2N_0} \sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \frac{1}{2N_0} \sum_{j=1}^{N_0} I_{0xj} = \frac{1}{2N_0} \left(\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \sum_{j=1}^{N_0} I_{0xj} \right)$$

Hence, the Oversampling classifier conditional probability of predicting $y_i = 1$ (positive for diabetes/prediabetes) given the feature vector $x_i = x'$ is

$$P_{ROS}(y = 1|x = x') = \frac{P_{ROS}((y = 1) \cap (x = x'))}{P_{ROS}(x = x')} =$$
$$= \frac{\frac{1}{2N_0} \sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj}}{\frac{1}{2N_0} \left(\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \sum_{j=1}^{N_0} I_{0xj} \right)} = \frac{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj}}{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} \, .$$

This yields Equation 8. □

Recalling that $t_{1j} \geq 0$ and comparing the ROS classifier conditional probability in Equation 8 with the imbalanced classifier conditional probability in Equation 7, it can be seen that

$$P_{ROS}(y = 1|x = x') = \frac{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj}}{\sum_{j=1}^{N_1} (t_{1j} + 1) I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} \geq \frac{\sum_{j=1}^{N_1} I_{1xj}}{\sum_{j=1}^{N_1} I_{1xj} + \sum_{j=1}^{N_0} I_{0xj}} = P(y = 1|x = x')$$

In fact, since ROS resamples multiple terms from the minority class (for which $t_{1j} > 0$), Equation 9 will yield $P_{ROS}(y = 1|x = x') > P(y = 1|x = x')$ if the feature vector x' is resampled from the minority class. This confirms that Random Oversampling will reduce

the problem of Type II (False Negative) error.

Appendix E: Systematic Resampling (Oversampling)

Now, we examine the impact of the Systematic Resampling (Oversampling) technique on the Bayesian classifier. Since this oversampling technique functions by conducting random sampling *with replacement* on k clusters from the minority class, we define a new *times count* function t_{1j} denoting the number of times that the j^{th} cluster C_j is resampled in the ROS Systematic Resampler, with $t_{1j} \geq 0$ and $\sum_{j=1}^{k} t_{1j} = k * \frac{N_0 - N_1}{N_1}$. However, since we also assume that the original minority class is included in the Oversampled dataset before any resampling is conducted, the j^{th} cluster C_j will appear a total of $t_{1j} + 1$ times. This also means that the element/feature vector $x_{1(j+qk)}$ from the minority class (for some $q \in \{0, 1, 2, ..., \frac{N_1}{k}\}$) appears $t_{1j} + 1$ times in the oversampled dataset.

As such, we take an indicator function I_{1jl} to denote whether feature vector x_{1jl} from the minority class fulfils the condition $x_{1jl} = x'$, in which case the total number of elements in the j^{th} cluster that fulfil this condition is $\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}$. Therefore, the probability of an element fulfilling the condition $x = x'$ in the j^{th} cluster is $P_j(x = x') = \frac{k}{N_1} \sum_{l=1}^{\frac{N_1}{k}} I_{1jl}$.

Hence, we determine that the total number of feature vectors that fulfill the condition $x = x'$ on the overall Systematic Resampled minority class is $\sum_{j=1}^{k} \left((t_{1j} + 1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl} \right)$, in which case the probability of $x = x'$ in the oversampled minority class is

$$P_{ROS}(x = x'|y = 1) = \frac{1}{N_0} \sum_{j=1}^{k} \left((t_{1j} + 1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl} \right)$$

As such, we determine that

$$P_{ROS}((y=1)\cap(x=x')) = P_{ROS}(x=x'|y=1) * P_{ROS}(y=1)\,,$$

Where we know that for the full Oversampled dataset (both Majority and resampled Minority classes) $P_{ROS}(y=1)=\frac{1}{2}$, since the Oversampled dataset is balanced. As such, we determine that

$$P_{ROS}((y=1)\cap(x=x')) = \frac{1}{2N_0}\sum_{j=1}^{k}\left((t_{1j}+1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)$$

Turning to the majority class of the Oversampled dataset to find $P_{ROS}((y=0)\cap(x=x'))$, we define an indicator function I_{0j} to denote whether the feature vector x_{0j} fulfils the condition $x_{0j}=x'$. As such, the probability of $x=x'$ in the majority class is $P_{ROS}(x=x'|y=0) = \frac{1}{N_0}\sum_{j=1}^{N_0} I_{0j}$. Knowing that for the full Oversampled dataset (both Majority and resampled Minority classes) $P_{ROS}(y=0)=\frac{1}{2}$, we therefore have

$$P_{ROS}((y=0)\cap(x=x')) = P_{ROS}(x=x'|y=0) * P_{ROS}(y=0) = \frac{1}{2N_0}\sum_{j=1}^{N_0} I_{0j}$$

Therefore, we determine that the full probability that a feature vector fulfils the condition $x=x'$ in the overall Oversampled dataset (including both Majority and Oversampled Minority classes) is

$$P_{ROS}(x=x') = P_{ROS}((y=1)\cap(x=x')) + P_{ROS}((y=0)\cap(x=x')) = \\ = \frac{1}{2N_0}\left(\sum_{j=1}^{k}\left((t_{1j}+1) * \sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right) + \sum_{j=1}^{N_0} I_{0j}\right)$$

As such, the Systematic Resampling classifier conditional probability is

$$P_{ROS}(y=1|x=x') = \frac{P_{ROS}((y=1)\cap(x=x'))}{P_{ROS}(x=x')} =$$
$$= \frac{\frac{1}{2N_0}\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)}{\frac{1}{2N_0}\left(\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)+\sum_{j=1}^{N_0} I_{0j}\right)} = \frac{\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)}{\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)+\sum_{j=1}^{N_0} I_{0j}}$$

This yields Equation 10. □

Since the original (imbalanced) dataset is equivalent to taking $t_{1j}=0, \forall j \in \{1,2,...,k\}$, we determine that $P(y=1|x=x') = \frac{\sum_{j=1}^{k}\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}}{\sum_{j=1}^{k}\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}+\sum_{j=1}^{N_0} I_{0j}}$, and we observe that $\sum_{j=1}^{k}\sum_{l=1}^{\frac{N_1}{k}} I_{1jl} = \sum_{j=1}^{N_1} I_{1xj}$ since each cluster is nonoverlapping by definition. Therefore, the imbalanced classifier conditional probability takes the form

$$P(y=1|x=x') = \frac{\sum_{j=1}^{N_1} I_{1xj}}{\sum_{j=1}^{N_1} I_{1xj}+\sum_{j=1}^{N_0} I_{0xj}}$$

Which matches Equation 7. □

Therefore, since for $t_{1j} \geq 0$, $\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right) \geq \sum_{j=1}^{N_1} I_{1xj}$, comparing Equation 10 (for the Systematic Resampled dataset) with Equation 7 (for imbalanced original dataset), it can be observed that

$$P_{ROS}(y=1|x=x') = \frac{\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)}{\sum_{j=1}^{k}\left((t_{1j}+1)*\sum_{l=1}^{\frac{N_1}{k}} I_{1jl}\right)+\sum_{j=1}^{N_0} I_{0j}} \geq$$
$$\geq \frac{\sum_{j=1}^{N_1} I_{1xj}}{\sum_{j=1}^{N_1} I_{1xj}+\sum_{j=1}^{N_0} I_{0xj}} = P(y=1|x=x')\,.$$

This yields the relation given in Equation 11. □

In fact, since ROS resamples multiple clusters from the minority class (for which

$t_{1j} > 0$), Equation 11 will yield $P_{ROS}(y = 1|x = x') > P(y = 1|x = x')$ if the j^{th} cluster is resampled from the minority class and contains at least one feature vector that fulfils the condition $x_{1jl} = x'$. This confirms that Systematic Resampling as a method of Random Oversampling will reduce the problem of Type II (False Negative) error in Transfer Learning classification.