Bias-Corrected Machine Learning Prediction for Non-random Surveys

by

Olena Kislenko

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Data Science

Waterloo, Ontario, Canada, 2025

© Olena Kislenko 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Tuberculosis (TB) remains a significant global health concern, particularly in low- and middle-income countries, where diagnostic resources are often limited. As computational tools become increasingly accessible, machine learning (ML) presents promising avenues to improve TB screening and diagnosis, particularly when only survey data are available. However, many of these surveys rely on non-probability sampling strategies, such as quota sampling, which can introduce substantial bias into predictive models. If not adequately accounted for, this bias may distort model performance and lead to inaccurate conclusions. In this thesis, we propose a simulation-based bias correction framework that utilizes auxiliary variables to more accurately represent the full population and enhance prediction reliability. Multiple classification methods were applied to real-world TB screening data from Nigeria, India, and Indonesia, and the performance of the methods was assessed both with and without bias correction.

The simulation results demonstrate that XGBoost and support vector machines with a radial kernel consistently outperform other models in terms of Matthews correlation coefficient (MCC) and F1 score. However, specificity remains low due to the class imbalance in the data, with actual negative cases being underrepresented. Among the auxiliary variables considered, *weight loss* produced the strongest predictive improvements, aligning with prior clinical findings. Neural networks, though occasionally competitive, exhibited greater variability across simulation runs and generally lower MCC values compared to the best-performing traditional ML models. Their higher computational demands and sensitivity to sample composition may limit the practicality of neural networks in smaller or imbalanced datasets. When the models were applied to real-life data, the relative performance of the predictive models shifted depending on the auxiliary variable used for correction. This variability emphasizes the importance of evaluating multiple corrected models in practical applications, as no single approach is universally optimal across all scenarios.

In conclusion, the thesis extends existing methodologies to classification problems and demonstrates their relevance in the diagnosis of TB. Our findings support the integration of bias correction into predictive modelling with non-random samples, especially in global health contexts where data imbalance and limited resources are common.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Zelalem Negeri, for his guidance, patience, and encouragement throughout my Master's program. I am also thankful to our collaborator, Professor Oga-Omenka, for her expert insight on the topic, and to my committee members, Professor Zhu and Professor Diao, for their valuable feedback and advice. I also wish to thank all the faculty members for their guidance, support, and dedication, which have greatly enriched my experience in the program.

I am truly grateful to my friends, both within and outside the department, for their constant support and belief in me. Most of all, I thank my family, whose unwavering encouragement and love made this achievement possible.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures vii

List of Tables xiii

1 Introduction 1

1.1 Background 1

1.2 TB Diagnosis 2

1.3 Bias Correction Approaches 4

1.4 Outline of the Thesis 5

2 Methodology 6

2.1 Participant Selection 6

2.2 Data Description 8

2.3 Statistical Methodology 9

2.3.1 Bias Correction Methods 9

2.3.2 Model Development 12

2.3.3 Model Evaluation 16

3 Results **18**

3.1 Prediction Results 18

3.2 Real-life Data Results 24

4 Discussions and Conclusions **32**

4.1 Discussions 32

4.2 Conclusions 35

References **36**

APPENDICES **42**

A Neural Network Models Architectures for Section 2.3.3. **43**

B Additional Simulation Results **49**

C The Boxplots of the Simulation Results **62**

D Additional Results from the Real-Life Data **87**

List of Figures

2.1 Plot of the correlations between variables and TB status of participants. . 11

2.2 The architecture of the first (base) model. 15

3.1 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable. 19

3.2 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable. 20

3.3 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable. 20

3.4 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable. 21

3.5 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable. 21

3.6 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables. 22

3.7 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable. 23

3.8 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable. 24

3.9 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable. 25

3.10 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable. 26

3.11 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable. 27

3.12 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable. 28

A.1 The architecture of the second (deeper) model. 44

A.2 The architecture of the third (smaller) model. 45

A.3 The architecture of the fourth (wider) model. 46

A.4 The architecture of the fifth model (with batch normalization). 47

A.5 The architecture of the fifth model (with LeakyReLU). 48

C.1 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable. 62

C.2 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable. . . . 63

C.3 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable. 63

C.4 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable. 64

C.5 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable. 64

C.6 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable. . . 65

C.7 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable. . . 65

C.8 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable. 66

C.9 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable. 66

C.10 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable. . . 67

C.11 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables. 67

C.12 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables. 68

C.13 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable. 69

C.14 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable. 69

C.15 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable. 70

C.16 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable. 70

C.17 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable. 71

C.18 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variable. 71

C.19 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data. 72

C.20 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data. 72

C.21 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data. 73

C.22 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data. 73

C.23 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data. 74

C.24 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data. 74

C.25 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data. 75

C.26 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data. 75

C.27 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data. 76

C.28 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data. 76

C.29 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data. 77

C.30 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data. 77

C.31 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data. 78

C.32 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data. 78

C.33 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data. 79

C.34 The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data. 79

C.35 The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data. 80

C.36 The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data. 80

C.37 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data. 81

C.38 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data. 81

C.39 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data. . . 82

C.40 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data. . . 82

C.41 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data. 83

C.42 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data. 83

C.43 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data. 84

C.44 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data. 84

C.45 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data. 85

C.46 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data. 85

C.47 The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variables and excluding them from the training data. 86

C.48 The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variables and excluding them from the training data. 86

List of Tables

2.1 Inclusion probabilities when both *weight loss* and *time passed* were auxiliary variables . . . 12

2.2 Typical Classification Table . . . 16

3.1 Results of the ML models applied to the real-life data without bias correction 25

3.2 Results of the neural networks applied to the real-life data without bias correction . . . 26

3.5 Results from the ML models on the real-life data with *weight loss* as an auxiliary variable . . . 28

3.3 Results of the ML models applied to the real-life data with *fever* as an auxiliary variable . . . 29

3.6 Results from the neural networks on the real-life data with *weight loss* as an auxiliary variable . . . 29

3.4 Results of the neural networks applied to the real-life data with *fever* as an auxiliary variable . . . 30

3.7 Training and evaluation time of different models (in seconds) . . . 30

B.1 Average results when considering *weight loss* as an auxiliary variable (oversample probability=0.61, N=1000) . . . 50

B.2 Average results from neural networks when considering *weight loss* as an auxiliary variable (oversample probability=0.61, N=1000) . . . 50

B.3 Average results when considering *time passed* as an auxiliary variable (N=1000) 51

B.4 Average results from neural networks when considering *time passed* as an auxiliary variable (N=1000) . . . 51

B.5 Average results when considering *fever* as an auxiliary variable (inclusion probability=0.67, N=1000) . . . 52
B.6 Average results from neural networks when considering *fever* as an auxiliary variable (oversample probability=0.67, N=1000) . . . 52
B.7 Average results when considering *night sweats* as an auxiliary variable (inclusion probability=0.34, N=1000) . . . 53
B.8 Average results from neural networks when considering *night sweats* as an auxiliary variable (oversample probability=0.34, N=1000) . . . 53
B.9 Average results when considering *lack of appetite* as an auxiliary variable (inclusion probability=0.34) . . . 54
B.10 Average results from neural networks when considering *lack of appetite* as an auxiliary variable (oversample probability=0.34, N=1000) . . . 54
B.11 Average results when considering *weight loss and time passed* as auxiliary variables (N=1000) . . . 55
B.12 Average results from neural networks when considering *weight loss and time passed* as auxiliary variables (N=1000) . . . 55
B.13 Average results when considering *weight loss* as an auxiliary variable and excluding it from the training data (oversample probability=0.61, N=1000) 56
B.14 Average results from neural networks when considering *weight loss* as an auxiliary variable and excluding it from the training data (oversample probability=0.61, N=1000) . . . 56
B.15 Average results when considering *time passed* as an auxiliary variable and excluding it from the training data (N=1000) . . . 57
B.16 Average results from neural networks when considering *time passed* as an auxiliary variable and excluding it from the training data (N=1000) . . . 57
B.17 Average results when considering *fever* as an auxiliary variable and excluding it from the training data (inclusion probability=0.67, N=1000) . . . 58
B.18 Average results from neural networks when considering *fever* as an auxiliary variable and excluding it from the training data (N=1000) . . . 58
B.19 Average results when considering *night sweats* as an auxiliary variable and excluding it from the training data (inclusion probability=0.34, N=1000) . 59

B.20 Average results from neural networks when considering *night sweats* as an auxiliary variable and excluding it from the training data (oversample probability=0.34, N=1000) . . . 59

B.21 Average results when considering *lack of appetite* as an auxiliary variable and excluding it from training data (inclusion probability=0.34) . . . 60

B.22 Average results from neural networks when considering *lack of appetite* as an auxiliary variable and excluding it from the training data (oversample probability=0.34, N=1000) . . . 60

B.23 Average results when considering *weight loss and time passed* as auxiliary variables and excluding them from the training data (N=1000) . . . 61

B.24 Average results from neural networks when considering *weight loss and time passed* as auxiliary variables and excluding them from the training data (N=1000) . . . 61

D.1 Results from the ML models on the real-life data with *night sweats* as an auxiliary variable . . . 87

D.2 Results from the neural networks on the real-life data with *night sweats* as an auxiliary variable . . . 88

D.3 Results from the ML models on the real-life data with *lack of appetite* as an auxiliary variable . . . 88

D.4 Results from the neural networks on the real-life data with *lack of appetite* as an auxiliary variable . . . 89

D.5 Results from the ML models on the real-life data with *time passed* as an auxiliary variable . . . 89

D.6 Results from the neural networks on the real-life data with *time passed* as an auxiliary variable . . . 90

D.7 Results from the ML models on the real-life data with *weight loss and time passed* as the auxiliary variables . . . 90

D.8 Results from the neural networks on the real-life data with *weight loss and time passed* as the auxiliary variables . . . 91

D.9 Most important features by auxiliary variable scenario . . . 91

Chapter 1

Introduction

1.1 Background

Tuberculosis (TB) is an infectious disease caused by Mycobacterium tuberculosis, mainly affecting the lungs (pulmonary TB) but also capable of impacting other parts of the body (extrapulmonary TB) [1]. People with TB can experience symptoms such as a severe cough (sometimes accompanied by mucus or blood), chest pain, loss of appetite, weight loss, fatigue, fever, and others [2]. These bacteria usually spread through the air when a person with TB speaks, coughs, or sneezes. TB remains a global health concern, especially in low- and middle-income countries where resources are limited. According to the World Health Organization (WHO), TB returned as the leading cause of death from a single infectious agent in 2023, surpassing COVID-19, which had held this position for the previous three years [1]. In that year alone, approximately 8.2 million new cases of TB were diagnosed, and an estimated 1.25 million people died from the disease. At the same time, only eight countries represented more than two-thirds of the total global TB diagnoses, including India (26%), Indonesia (10%), and Nigeria (4.6%).

The literature indicates that some factors are associated with having TB or developing complications. For example, there is some evidence that the people with comorbidities such as type 2 diabetes mellitus, chronic kidney disease, rheumatoid arthritis and HIV are at risk of acquiring TB [3, 4]. Also, Helle et al. (2024) showed that comorbidities increase the chances of death in people with and without TB in Central India [5]. The WHO mentions smoking, alcohol and substance use disorders, malnourishment, pregnancy and immunocompromising conditions such as organ transplant, renal failure and dialysis to be the risk factors for TB [6]. According to the guidelines, people with such risk factors may

be prioritized for TB screening. In addition, TB risk and outcomes vary by demographic factors. For instance, men are more likely to develop TB than women, with 6.0 million reported cases among adult men compared to 3.6 million among adult women [1].

Drug-resistant TB remains a major global health concern due to its complexity in diagnosis, longer and less effective treatment, and lower success rates compared to drug-susceptible TB [7]. It often arises from incomplete or improper treatment and is more difficult to manage, especially in resource-limited settings. Early detection is crucial for controlling its spread and improving outcomes.

1.2 TB Diagnosis

In this section, we will discuss the diagnosis of TB. Three of the main characteristics of a test are the accuracy, sensitivity (true positive rate) and specificity (true negative rate). One of the key features of TB diagnosis is that the sensitivity should be extremely high for the tool to be implemented in practice, since misclassifying the sick person as healthy could lead to serious consequences. On the other hand, incorrectly identifying a healthy person as diseased, although it is unwanted, might not be as crucial, as one could introduce further examinations to support the test's result.

Detection and timely treatment of TB prevents up to 50% mortality, while delays worsen severity, transmission, resistance, and burden of healthcare [8]. In 2015, the WHO proposed a strategy to end TB [9]. In particular, it emphasized the essential role of early diagnosis and systematic screening of contacts and high-risk groups. Although some progress has been made, it has not been sufficient to meet the first milestone of this strategy. TB can be diagnosed with a skin test or a blood test. However, it usually takes several days to obtain the results of these tests. Several rapid tests have been developed to overcome this drawback, including Xpert MTB/RIF, Xpert Ultra, Truenat MTB, Truenat MTB Plus, TB-LAMP and others [10]. Among these tests, GeneXpert (Xpert MTB/RIF) is one of the most widely used. However, the cost of this test, along with some operational issues, might raise concerns. It requires specialized equipment and skilled personnel, which may be inaccessible in low-income areas. A recent study found that the sensitivity of Gene Xpert for diagnosing spinal tuberculosis is 91.18%, while the specificity is 100% [11].

One of the recent advances is the use of machine learning (ML) tools to help solve healthcare-related questions [12]. ML can be used for various tasks, including solving hospital-level problems, such as predicting the next year's burden to estimate the required number of practitioners. Additionally, using ML technologies for personalized care is an

actively developing field. For example, predicting the risk, diagnosis, treatment monitoring or patient outcomes prediction are still active research areas. Regarding personalized care for TB, some advancements have already been made [13].

Several tools for automatic TB diagnosis have been proposed previously [14, 15]. In particular, the detection of pulmonary TB based on chest X-rays has been extensively studied. The first such neural network was created in 1999 by El-Solh et al. [16]. They developed a General Regression Neural Network (GRNN) to predict TB, achieving a sensitivity of 100% and a specificity of 69% in new patients. Zihan et al. (2024) created the Convolutional Neural Network (CNN) with the EfficientNet-B0 architecture trained on three datasets: the Tuberculosis Chest X-ray Database, the Shenzhen Hospital CXR Set, and the Montgomery County CXR Set[17]. The reported accuracy on the combined data is 98.50%. However, an important drawback of using X-rays for the TB diagnosis is that it cannot exclude extrapulmonary TB. For example, in the case of skeletal TB, also known as bone TB, computed tomographic (CT) scans and magnetic resonance imaging (MRI) are recommended, as they are more sensitive [18, 19].

Researchers have also made some progress with developing ML approaches for data other than X-rays. Elveren and Yumuşak (2011) trained the Artificial Neural Network with Genetic Algorithm [20]. They used participants' epicrisis reports from Diyarbakir Chest Disease Hospital. These reports included data on symptoms the participants experienced as well as laboratory test results, including erythrocyte (RBC), hemoglobin (HGB), thrombocyte (PLT), cholesterol, creatinine and others. The reported accuracy of their detection algorithm is 94.88%. Barros et al. (2021) explored the Brazilian Information System for Notifiable Diseases (SINAN) database, which includes socio-demographic, clinical, and laboratory data on TB patients from Brazil [21]. They aimed to predict the probability of death in TB patients, and achieved the best results with the Gradient Boosting and an ensemble model composed of Random Forest, Gradient Boosting, and Multi-Layer Perceptron. Their Gradient Boosting model yielded a sensitivity and specificity of 97.22% and 95.91%, respectively, when applied to the balanced dataset. At the same time, the ensemble model achieved 97.12% and 95.74% in sensitivity and specificity, respectively. However, their study was limited by the use of data from a single region and significant class imbalance, which they addressed through random under-sampling. This approach may reduce data representativeness and affect generalizability, highlighting the need for bias correction methods and validation on more diverse datasets. Karmani et al. (2024) conducted a survey in Pakistan and then applied and compared Decision Tree, Gaussian Naive Bayes, Logistic Regression, Adaptive Boosting and Neural Network to diagnose TB in the participants [22]. AdaBoost achieved an accuracy of 93.42%. In addition, Septiandri et al. (2019) used the verbal screening questionnaire and cost-sensitive training to diagnose

TB [23]. However, these existing ML approaches assume that the data are independent and identically distributed (iid) or come from a probability sampling survey. The following section discusses some proposed solutions to overcome this limitation.

1.3 Bias Correction Approaches

One of the gaps in the studies using computer-aided diagnosis is the presence of selection bias. Usually, the analysis assumes random sampling, which is not always true. One of the standard sampling techniques is quota sampling, where the population is divided into subgroups and the participants are selected from each subgroup until the desired quota is filled. For example, quota sampling is used in a setting where several hospitals are involved in data collection, and there is a maximum number of participants whose data are collected from each hospital. Further data analysis would require incorporating a bias correction technique in these settings. Ignoring selection bias may cause models to perform well on the sampled data but poorly when applied to other populations.

Gutierrez-Velez et al. (2020) proposed a method to detect and mitigate spatial bias when modelling species occurrence [24]. Later, Liu et al. (2021) proposed a spatial weighted tree method to deal with spatial bias [25]. In [26], the authors proposed a model of sampling bias called conditional Γ-biased sampling and an approach to learning the decision rules under Γ-biased sampling.

Some methods were also developed to estimate the population mean. Rafei et al. (2022) proposed a fully model-based method for inference using non-probability samples where the goal is to predict the outcome variable for the entire population unit [27]. Similarly, Liu et al. (2023) propose Bayesian machine learning model-based methods for estimating population quantities from nonrandom samples in settings where high-dimensional auxiliary information is available in the sample and the target population. [28]

In addition, Artificial Intelligence (AI) systems have also been found to be prone to bias [29]. A well-known example is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS). COMPAS is a risk-assessment tool developed for predicting a defendant's risk of committing another crime based on responses to 137 questions. It has been revealed that this algorithm exhibited racial bias, even though race was not explicitly included in the questionnaire [30]. This and other similar examples led to the development of the literature on the problem of unfairness in ML. Kamishima et al. (2012) discussed three possible causes of bias in classification tasks (prejudice, underestimation, and negative legacy) and proposed a regularization approach to remove prejudice [31]. In

[32], the authors proposed a general framework for learning fair classifiers that practitioners can directly incorporate into classification algorithms. Additionally, Mehrabi et al. (2021) designed an attention-based model, which led to a post-processing bias mitigation strategy [33].

Additionally, Buelens, Burger and van den Brakel (2015) proposed a bias-correction method for non-probability samples and demonstrated it with a simulation study [34]. However, their report assumes a continuous outcome and does not address selection approaches for auxiliary variables and the best model. This study will use a similar idea but for the binary outcome, which, to the best of our knowledge, has not been addressed yet. Therefore, the objectives of this study are to (1) propose and evaluate novel computational methods to correct selective bias and (2) to use the corrected data to predict TB diagnosis more accurately.

To achieve these objectives, we develop a simulation-based framework that treats the original dataset as population data, from which biased samples are generated by overrepresenting specific subgroups based on auxiliary variables associated with TB diagnosis. These samples mimic real-life nonprobability sampling scenarios. We then apply and compare a range of ML algorithms – including logistic regression, random forests, XGBoost, support vector machines, and neural networks – both with and without bias correction. The performance of the models is evaluated using sensitivity, specificity, accuracy, balanced accuracy, Matthews Correlation Coefficient (MCC), and F1 score to identify the most effective approach. Ultimately, we compare corrected and uncorrected models.

1.4 Outline of the Thesis

Chapter 2 describes the methodology of this work, including an explanation of the participant selection process, a description of the data, the proposed bias correction approach, and a background on the ML methods used. Chapter 3 presents the prediction results obtained from the simulations and the real-life data. Finally, Chapter 4 provides a summary of the findings, a discussion of the implications, an examination of the limitations and future directions, and a conclusion.

Chapter 2

Methodology

2.1 Participant Selection

This study will use secondary data collected from surveys conducted in India, Indonesia and Nigeria [35, 36]. We want to point out that the participant selection criteria differed slightly between the three surveys.

In Nigeria, the data was collected by the survey firm Hanovia Inc. They randomly selected a number of provider sites, stratified by geographical area, past TB notifications and type of provider, that were willing to be a research site. The recruitment facilities were selected across the states. The higher volume facilities were prioritized for selection. As a result, nine recruitment facilities were chosen from nine different local government areas (LGAs) in each state for 18 recruitment sites. However, one of the facilities did not agree to participate, so it was replaced with another high-volume facility in the same senatorial district.

Within each recruitment facility, Hanovia's staff worked with SHOPS Plus and facility staff to recruit ten study participants per facility to respond to the quantitative surveys. The participants selected for the surveys were people with chest symptoms confirmed negative for TB or people with confirmed TB. Two exclusion factors were age and HIV status: The study participants recruited had to be adults over 18 years of age and HIV-negative. For chest symptomatic participants, the team confirmed that the person had a negative TB sputum test on the TB register. For confirmed TB participants, the team used the TB treatment register to select the most recent patient to be placed on treatment at the facility and not transferred from a different facility. In both cases, the participants

needed to have interacted with the facility no earlier than January 1, 2021, to minimize recall bias. Recruitment in each facility stopped when the recruitment quota for that facility was reached or when a person diagnosed with TB or identified as presumptive for TB before January 1, 2021, was reached (whichever milestone came first).

Starting with the most recently-registered patient, potential study participants were contacted about the study were informed about the purpose of the study, asked to indicate their initial willingness to participate, and asked to set a date and time and place (the clinical facility, the patient's home, or place of work) to administer the survey. Patients provided initial consent to participate in the study during recruitment and would then be formally consented again, before commencing their interviews or surveys. Approximately 10-15% of all consenting patients, selected based on age group, sex, occupation, marital status, and geographical location, were asked to participate in a detailed qualitative interview instead of tablet-based questionnaires. This thesis will consider the answers collected only from quantitative surveys. The questionnaires were translated into the Hausa and Yoruba languages and the participants could choose which language to use for the survey. Overall, the aim was to recruit 5-7 chest symptomatics and 5-7 TB participants per facility for the state threshold of 90 complete surveys. However, in most cases, facilities tried to recruit 1-2 more participants over their quota to have a backup sample.

In India, only participants with confirmed tuberculosis (TB) were included in the study, which was conducted between February and March 2022 in two cities: Mumbai and Patna. A total of 400 participants were recruited, with 200 from each city. Trained field officers carried out recruitment from the Institute for Social and Economic Research on Development and Democracy (ISERDD). Eligible participants were adults (aged 18 or older) who had either sought care for TB-related symptoms or received a TB diagnosis during the COVID-19 pandemic. Purposive sampling was employed to ensure representation of diverse care-seeking experiences, including individuals who accessed both public and private healthcare services. Data collection took place through in-person interviews at TB treatment centers, using tablet-based electronic forms.

In Indonesia, the study occurred in a randomly chosen area in Bandung. The recruitment facilities in this area were selected to have patients who potentially meet the inclusion criteria and are willing to participate in the study. Participants had to be over 18 years old with confirmed pulmonary TB or some respiratory symptoms (no more than 3 months after diagnosis) but with negative results of the TB test. People who had received some TB treatment six months before the data collection were excluded. Additionally, patients diagnosed with TB without a standardized sputum test were excluded. The study involved 59 private practitioners or clinics, 10 community health care centers, 3 public hospitals, and four private hospitals distributed in 36 community health centers in the chosen area.

The quota on the number of study participants to be interviewed in each health facility was predetermined. The target was to recruit 150 participants with TB and 50 with respiratory disease from private health facilities, and 100 participants with TB from public health facilities. The questionnaires were translated into Bahasa Indonesia, Indonesia's national language.

2.2 Data Description

A total of 180 study participants from Nigeria, 396 from India and 469 from Indonesia were surveyed. The non-probability sampling technique (quota sampling) was used when choosing the study participants, which can lead to biased estimates and predictions if not adjusted for due to unrepresentative samples.

The common features for the three countries that this thesis will use to make predictions are the following:

- Outcome variable: TB status.
- Demographic information: gender, marital status, age split into three groups – youth (18-24 years old), adults (25-64 years old) and seniors (over 65).
- Symptoms: cough, sore throat, weight loss, night sweats, fever, and lack of appetite. Each feature is a binary variable indicating whether a participant experienced this symptom.
- Participant's HIV status.
- TB contact: This variable indicates whether the participant had had contact with those diagnosed with TB.
- Time passed: This variable indicates the amount of time since the participant started experiencing symptoms. It is split into the following levels: 2 days or less, 3-7 days (up to 1 week), more than 1 week but less than 1 month, 1-2 months, 2-3 months, and over 3 months. This variable is numbered from 1 to 6, where 1 corresponds to '2 days or less' and 6 to 'over 3 months'.

Age will be taken as the age of study participants at which the questionnaire was completed. For the Indonesian dataset, exact survey dates were not available. However, it

is known that they were conducted between July 14, 2021 and February 24, 2022. We will assume that the age of the participants from Indonesia is equal to the difference between 2021 and the participant's birth year. Furthermore, in the Indian data, only the birth year is known instead of the exact date of birth. In this case, we will consider that the age at the interview is equal to the difference between the year of the interview and the birth year.

In the Indian study, not all participants were adults (over 18 years of age), but we will only consider adults in this study. Therefore, 8 participants (6 from Mumbai and 2 from Patna) who were under the age of 18 were excluded. Furthermore, some participants in the Indonesian study had inconsistent or unreliable TB test results, and such participants were also excluded. Since all the participants with definite test results had a positive test result, only people with TB were considered from the Indonesian study. In addition, three study participants in the Indian survey had missing TB status. However, based on the participant selection criteria, we replaced their statuses with a value of 1, assuming that these participants have been diagnosed with TB. In general, surveys from 816 people (180 from Nigeria, 388 from India and 248 from Indonesia) were used for this study.

Since having TB was the inclusion criterion for the Indian and Indonesian studies, only 90 participants from the Nigerian survey out of 816 total were not diagnosed with TB. This causes a class imbalance, which might influence trained ML models.

First, we will prepare the data before applying the ML algorithms. This includes using one-hot encoding to handle categorical features such as gender, where each category is represented as a separate binary feature. Some variables, such as the participant's HIV status, have many missing values. We will be treating the missing values as a separate category. When performing one-hot encoding, one column corresponding to each feature with missing values present will be an indicator of whether the value is absent. In addition, we will randomly split the data into training and test subsets to later evaluate the trained ML models on the unseen test data.

2.3 Statistical Methodology

2.3.1 Bias Correction Methods

As discussed in the previous section, quota sampling was used to collect data in the three surveys, which can introduce bias into predictions based on this nonprobability sampling

technique. Therefore, this section will describe our proposed simulation-based bias correction technique in detail.

First, we will treat our original data as *population* data. We will create several scenarios by taking samples with replacement from the data. We will identify auxiliary variables as the participants' characteristics that are highly correlated with being diagnosed with TB (for example, age, gender, or symptoms). Because these variables are closely related to the outcome, they serve as effective proxies for the selection process, enabling more accurate bias correction. Second, we introduce selective bias to our samples by using the identified auxiliary variables to generate samples with replacement from the population data that mimics real-life data creation scenarios. We do this by overrepresenting a part of the population by increasing the inclusion probability of one auxiliary variable at a time. For example, if older people are more likely to be diagnosed with TB, we will sample more older people than younger ones. The inclusion probability for each value of the auxiliary variable is determined as the proportion of participants in the population who share that value. When calculating the inclusion probabilities for the continuous variables, such as age, we will treat them as categorical by introducing a cut-off value. To create some scenarios, we will also consider multiple auxiliary variables at a time, such as overrepresenting females with coughs. This way, the sampled scenarios will generate nonrepresentative samples that resemble different real-life nonprobability-based samples. For each sample, we will then apply the ML algorithms while including and excluding the auxiliary variables used to generate the sample, and compare the predictions on the test set for this sample. This step will help us evaluate the performance of the ML algorithms with or without bias correction.

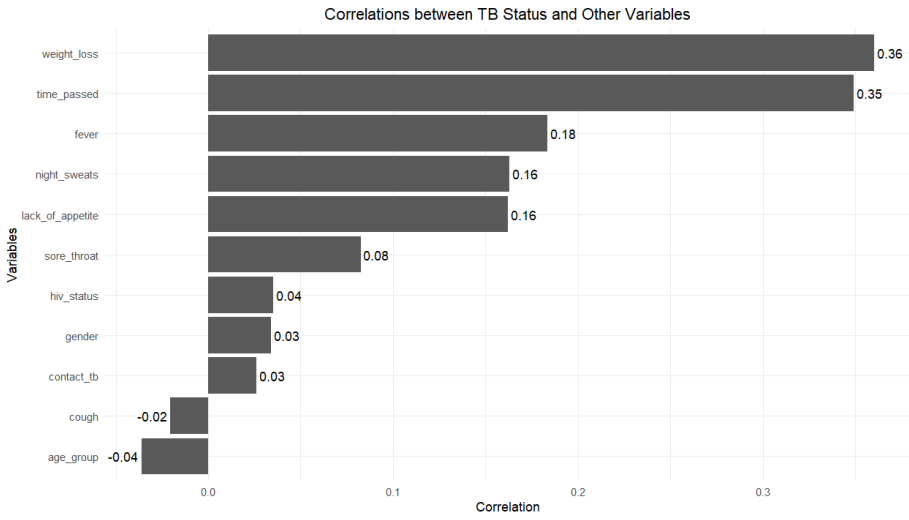


Figure 2.1: Plot of the correlations between variables and TB status of participants.

In our case, we have identified the auxiliary variables based on Spearman's correlations between each ordinal variable and the TB status of the participants (our response variable). These correlations are presented in Figure 2.1. There were no strong (0.75+) or moderate (0.5+) correlations. However, *weight loss* and *time passed* are the most associated with TB. We also considered *fever*, *night sweats* and *lack of appetite* as possible auxiliary variables. In addition, we will also consider a scenario in which two variables with the strongest correlations (*weight loss* and *time passed*) are treated as auxiliary variables.

In four out of six scenarios, only one binary variable was chosen as an auxiliary variable. In these cases, when overrepresenting a part of the population, we will use the following inclusion probabilities for participants with auxiliary variables' value equal to 1: 0.61 for *weight loss*, 0.67 for *fever*, 0.34 for *night sweats* and 0.34 for *lack of appetite*. For *time passed*, which has six possible levels numbered from 1 to 6, we use the following inclusion probabilities: 0.073, 0.158, 0.23, 0.213, 0.116, and 0.209 for levels 1 to 6, respectively. Lastly, the inclusion probabilities for scenarios in which both *weight loss* and *time passed* were auxiliary variables are presented in Table 2.1. These inclusion probabilities were calculated based on the distributions of the auxiliary variables.

Table 2.1: Inclusion probabilities when both *weight loss* and *time passed* were auxiliary variables

		Time Passed					
Weight Loss		**1**	**2**	**3**	**4**	**5**	**6**
	0	0.05	0.10	0.10	0.07	0.03	0.04
	1	0.03	0.06	0.13	0.14	0.09	0.16

2.3.2 Model Development

Classic Machine Learning Models

In the model development stage, we will apply advanced ML algorithms to construct predictive models capable of detecting TB cases. These will include logistic regression[37], XGBoost (eXtreme Gradient Boosting)[38], random forests[39], and support vector machines (SVMs)[40].

Logistic regression will be used as a simple baseline model to establish a reference for comparison with more complex algorithms. Logistic regression uses the linear regression model and transforms its outcome into a value between 0 and 1, representing probability. Mathematically, given a feature vector $x = (x_1, x_2, ..., x_d)$, logistic regression models the probability of TB presence ($Y = 1$) as:

$$P(Y = 1|x) = p(x) = \frac{1}{1 + e^{-z}},$$

where z is obtained after a linear transformation of the input vector $x = (x_1, ..., x_d)$:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_d x_d.$$

Here, parameters $\beta_0, \beta_1, ..., \beta_d$ are the regression coefficients that can be estimated via maximum likelihood. The cost function that needs to be minimized is the negative log likelihood:

$$l = -\sum_{i=1}^{N} \left[y_i log(p_i) + (1 - y_i) log(1 - p_i) \right],$$

where N is the number of observations in the training set and for given x_i and y_i, $p_i = p(x_i)$.

Random forests and XGBoost, known for their robustness and ability to handle complex data structures, will capture non-linear relationships and interactions within the dataset.

Both models are based on decision trees[41]. The decision tree is a recursive model in which each internal node represents a decision rule based on a single feature that divides the dataset into smaller subsets. Random forests construct a set of decision trees. At each training step, only a random subset of features is chosen for a potential split. This diversifies the trees by reducing the correlation between them. If there is a total of d features, using $\sqrt{d}$ features for every split is common. For every observation x, every tree in the ensemble produces a class probability, and the final probability is obtained by averaging the received probabilities:

$$\hat{P}(Y = 1|x) = \frac{1}{K} \sum_{k=1}^{K} P_k(Y = 1|x),$$

where K is the total number of trees and P_k is the probability estimated by the tree number k. XGBoost improves this model by using gradient boosting, where trees are added sequentially to correct the errors of the previous models, based on the gradient of the loss function for the binary classification task.

In addition, SVMs will be explored for their efficiency when working in high-dimensional feature spaces. SVMs seek to find the optimal hyperplane that maximizes the margin between the closest points of different classes. Given the training set of points (x_i, y_i) with $y_i \in \{-1, 1\}$, the optimization problem for the (soft margin) SVMs is the following:

$$\min_{w,b,\gamma} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \gamma_i$$

subject to the constraints:

$$y_i(w^T x_i + b) \geq 1 - \gamma_i \text{ and } \gamma_i \geq 0,$$

where w is the vector normal to the hyperplane, b is the offset bias term, C is a regularization parameter and γ_i are slack variables that allow some other points to be on the wrong side of the margin (in other words, they allow the data set to be non linearly separable). In some cases, it is beneficial to use the kernel function $K(x_i, x_j)$, such as a polynomial, radial, or Gaussian kernel, to transform the data, allowing SVMs to learn more complex decision boundaries. Our study will apply SVM with a linear and a radial kernel.

Lastly, the ensemble models[42] will be employed to capture the benefits of several models at a time. They combine the predictions of multiple models by aggregating their outputs, typically through majority voting. This approach helps to reduce individual model errors and improve overall prediction accuracy by leveraging the strengths of each model.

To ensure generalizability and account for issues related to high-dimensional data, such as model overfitting, regularization techniques such as Lasso[43], Ridge regression[44] and elastic net[45] will be applied. Lasso regularization (or L_1 regularization) introduces to the log-likelihood function an additional penalty term $L_1 = \lambda \sum_{i=1}^{d} |\beta_i|$. In contrast, Ridge regression (or L_2 regularization) uses the penalty of the following form: $L_2 = \lambda \sum_{i=1}^{d} \beta_i^2$. The elastic net combines both of these terms so that the regularization terms are the following: $\lambda_1 \sum_{i=1}^{d} |\beta_i| + \lambda_2 \sum_{i=1}^{d} \beta_i^2$. In this thesis, the three regularization techniques will be applied to logistic regression and XGBoost.

Throughout the hyperparameter choice process, we will employ k-fold cross-validation[46]. This technique randomly splits the data into k groups, trains the model in $k-1$ groups, tests the remaining one, and then repeats this process for each of the k groups. The model can be evaluated by averaging all k received performance metric values.

Furthermore, a feature importance analysis[39, 47] will be performed to identify the most informative factors that contribute to the models' predictive performance. This analysis will provide valuable information on the underlying mechanisms that drive TB and will help indicate key factors that influence disease progression.

Neural Networks

In addition to the classic ML methods described above, neural networks will be applied.

We will use six different models for this task. Each of them is a fully connected feed-forward neural network implemented using Keras. The architecture of the first ("base") neural network (see Figure 2.2) consists of an input layer followed by two hidden layers with 32 and 16 neurons, respectively, each using the ReLU activation function. Dropout layers with a rate of 0.2 are added after each hidden layer to prevent overfitting. The output layer contains a single neuron with a sigmoid activation function to produce a probability score for binary classification.

The second model is a deeper neural network with three hidden layers instead of two. It starts with 64 neurons in the first layer, followed by 32 and 16 neurons. Dropout layers with a slightly higher rate of 0.3 are placed after the first two hidden layers. Like the base model, it uses ReLU activations in the hidden layers and a sigmoid function in the output layer.

The third model is a smaller neural network with fewer parameters. It contains two hidden layers with 16 and 8 neurons and omits dropout layers. This design serves as a comparison to assess the performance of a more compact architecture.

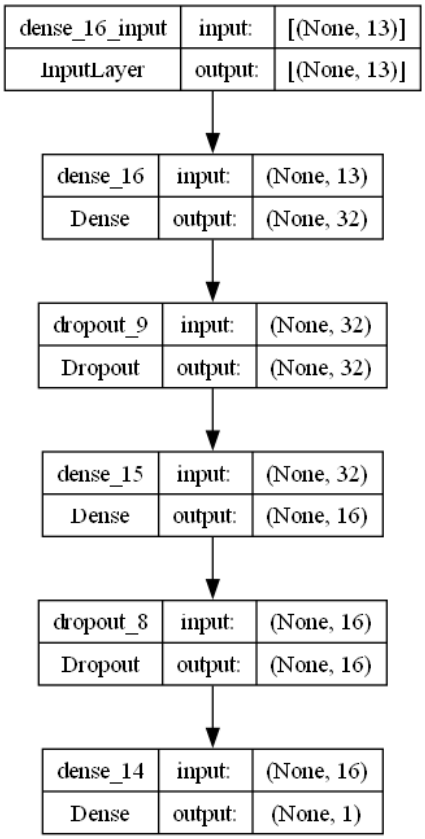


Figure 2.2: The architecture of the first (base) model.

The fourth model is a wider network that increases the number of neurons in each hidden layer. It uses 128, 64, and 32 neurons across three hidden layers, with dropout layers (rate = 0.3) after the first and second hidden layers. This model tests whether increasing network capacity improves performance.

The fifth model builds on the base architecture, but includes batch normalization layers after each hidden layer instead of dropout. Batch normalization can help stabilize and speed up training by normalizing activations within each mini-batch.

Finally, the sixth model also follows the base structure but replaces the ReLU activations with LeakyReLU. This variant of ReLU allows for a slight non-zero gradient when the unit is inactive, which may help address the 'dying ReLU' problem in some cases.

The architectures of the remaining neural network models can also be found in the appendix corresponding to these figures. Given the limited size of the training dataset, we

restrict our models to relatively small neural networks to reduce the risk of overfitting.

2.3.3 Model Evaluation

In the model evaluation phase, we will rigorously assess the performance of our predictive models to ensure their accuracy, robustness, and generalizability to detect TB cases.

First, we will explore and compare performance metrics to quantify the effectiveness of our models. Our main metrics in this study are MCC and F1 score, which are known for their relevance when the data is imbalanced. MCC, which is also referred to as the phi coefficient, is equal to Spearman's correlation coefficient between the actual and predicted data. It is considered to be a particularly reliable metric since it utilizes all four elements of the classification table [48]. Sensitivity, specificity, accuracy and balanced accuracy will also be calculated.

Table 2.2: Typical Classification Table

		True TB Status	
		Positive	**Negative**
Test Result	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Given the classification table in Table 2.2 corresponding to the model's performance in a given sample, we can calculate the performance metrics as follows: Sensitivity: $\frac{TP}{TP+FN}$, Specificity: $\frac{TN}{FP+TN}$, Accuracy: $\frac{TP+TN}{TP+FP+FN+TN}$, Balanced accuracy: $\frac{Sensitivity+Specificity}{2}$,

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

where

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}.$$

To evaluate and choose the best ML algorithm, we will calculate the average MCC of the methods in all simulated samples, and compare the prediction error of each model for the test set[49].

By systematically comparing the performance of the models received from each ML algorithm, we aim to identify the most effective and robust approach for the early detection of TB cases. This comparative analysis will provide valuable insight into each algorithm's relative strengths and weaknesses and inform the selection of the optimal model for deployment.

Once an effective and robust ML algorithm has been developed, we will compare its performance on datasets with and without bias correction. Finally, the validated ML algorithm can be used in practice to generate individualized predictions of TB diagnosis for participants with characteristics similar to those in the training data.

We conducted all analyses using the R[50] programming language versions 4.4.1 and 4.3.2 and RStudio[51] version 2023.12.0.

Chapter 3

Results

This chapter presents the results of applying the methodology described in the previous chapter to TB data from Nigeria, India, and Indonesia, as discussed in Chapter 1.

3.1 Prediction Results

Selected boxplots depicting the distributions of MCC prediction values and F1 scores, obtained from ML models trained on 1000 replicated samples, are presented in Figures 3.1–3.6. Boxplots of prediction sensitivities, specificities, accuracies, and balanced accuracies are provided in Appendix C. Additionally, Appendix B presents tables showing the average values of all the evaluated metrics for each method used.

As shown in the tables in Appendix B, an important observation in all scenarios, for all auxiliary variables, is that the average specificities are consistently much lower than the sensitivities or accuracies, indicating a high number of FPs. This also contributed to the low MCC values, which ranged from 0.469 to 0.794 across all methods and scenarios. This outcome is expected given the substantial class imbalance in the data. Only 90 of 816 (11%) participants were not diagnosed with TB. We did not address this imbalance directly, as doing so could have interfered with the proposed bias correction technique. Furthermore, the boxplots reveal considerably greater specificity variability across different sampled datasets. In contrast, based on the boxplots presented of F1 scores, sensitivities, and accuracies, the models' prediction qualities were consistently high with precise interquartile range (IQR), except for one scenario where logistic regression with Lasso regularization produced a wider boxplot for sensitivity. This indicates that the models were

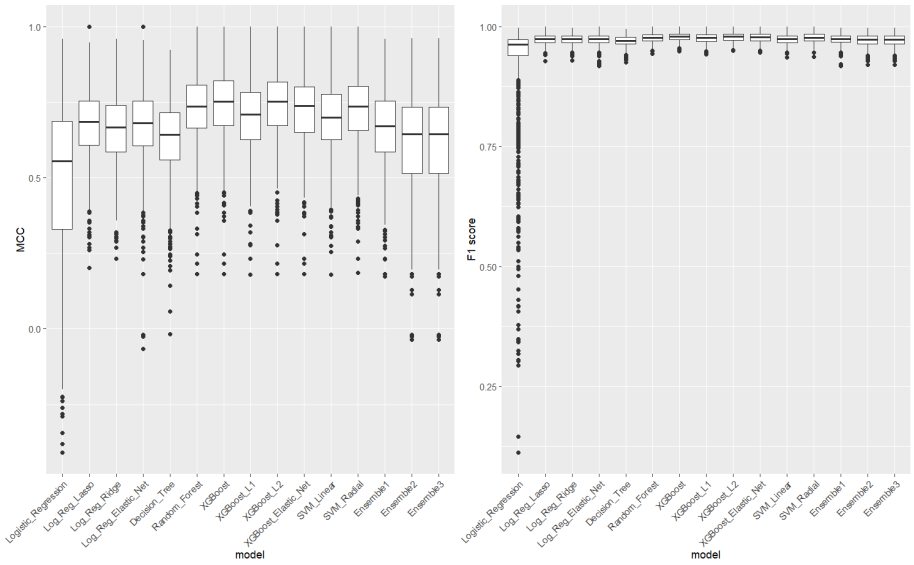


Figure 3.1: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable.

robustly predicting positive TB cases. Matthews correlation coefficient, which accounts for all elements of the confusion matrix, showed moderate variability. It was more stable than specificity, but less consistent than accuracy or sensitivity. However, given the class imbalance in the data, MCC provides a more reliable overall assessment of the performance of ML models.

The performance of the algorithms remained consistent across the different scenarios, with XGBoost, random forest, and SVM with a radial kernel demonstrating the strongest evaluation metrics, based on their MCC and F1 score values. Specifically, the average MCC values of XGBoost ranged from 0.731 to 0.794, while those of the random forest and radial SVM achieved slightly lower values, ranging from 0.711 to 0.777 and from 0.711 to 0.771, respectively (see Appendix B). Although the regularized versions of XGBoost produced results comparable to the base model, their added complexity makes the simpler variant preferable. Regularization did not noticeably improve performance, probably due to the limited dataset size. The introduction of additional parameters through regularization may require larger sample sizes to yield benefits. Additionally, since both random forest and XGBoost are tree-based models and XGBoost consistently performs better, it may not be meaningful to consider random forest among the top-performing methods.

Compared to other ML algorithms, logistic regression variants appeared to be more

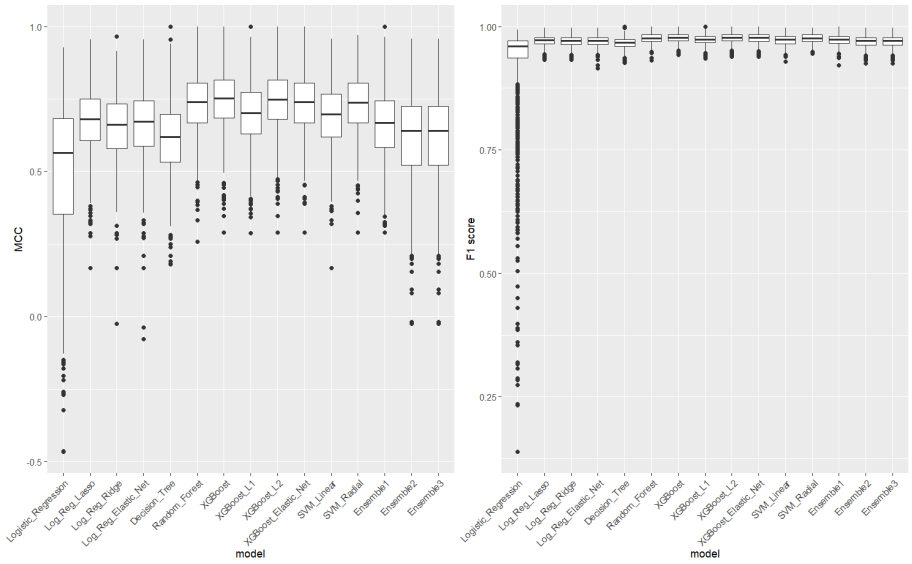


Figure 3.2: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable.

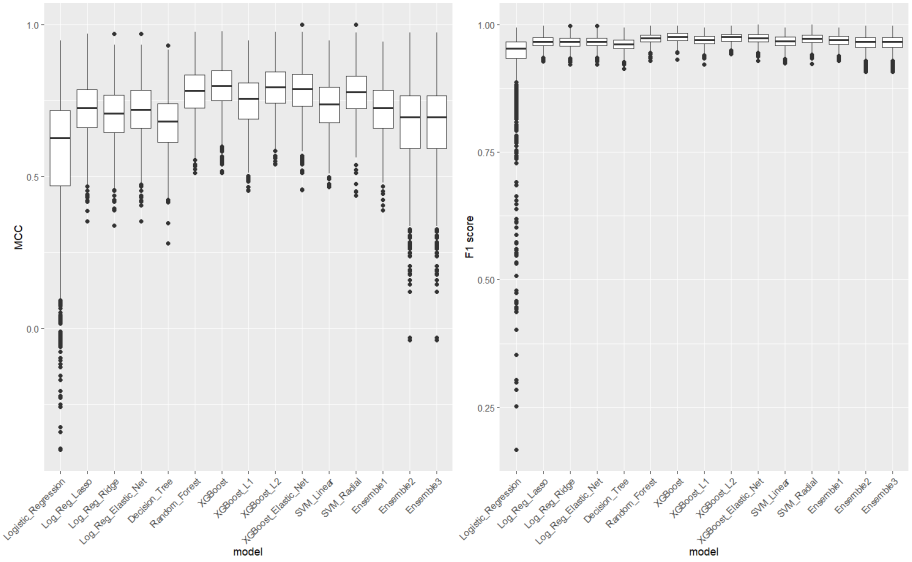


Figure 3.3: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable.

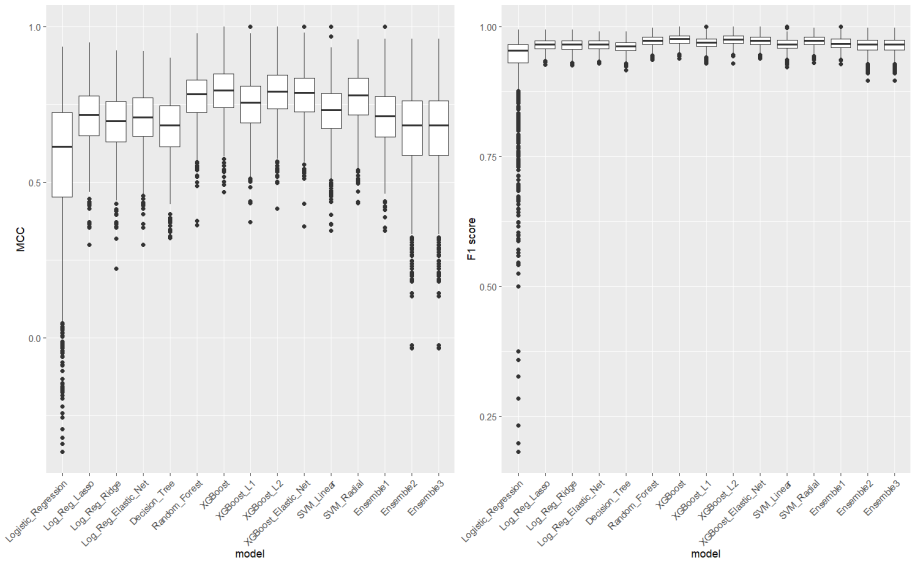


Figure 3.4: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable.

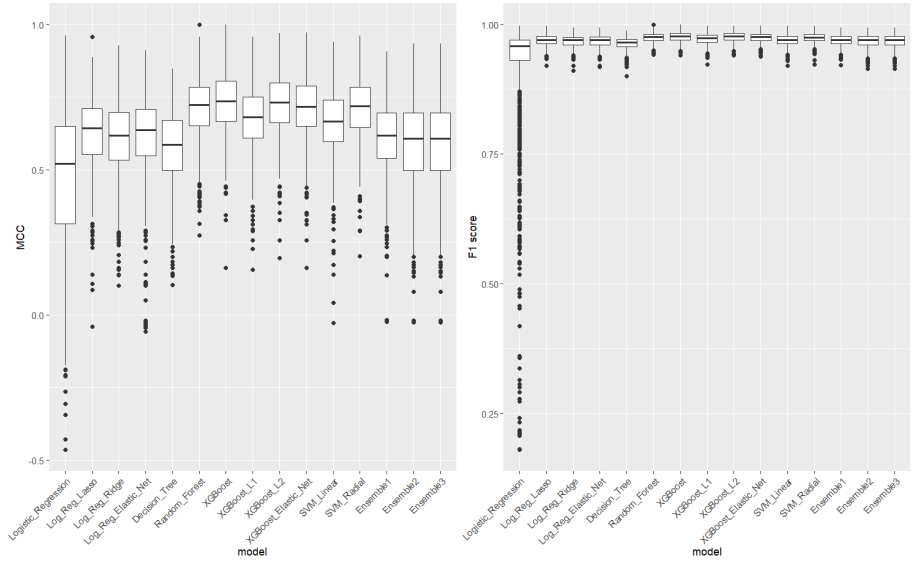


Figure 3.5: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

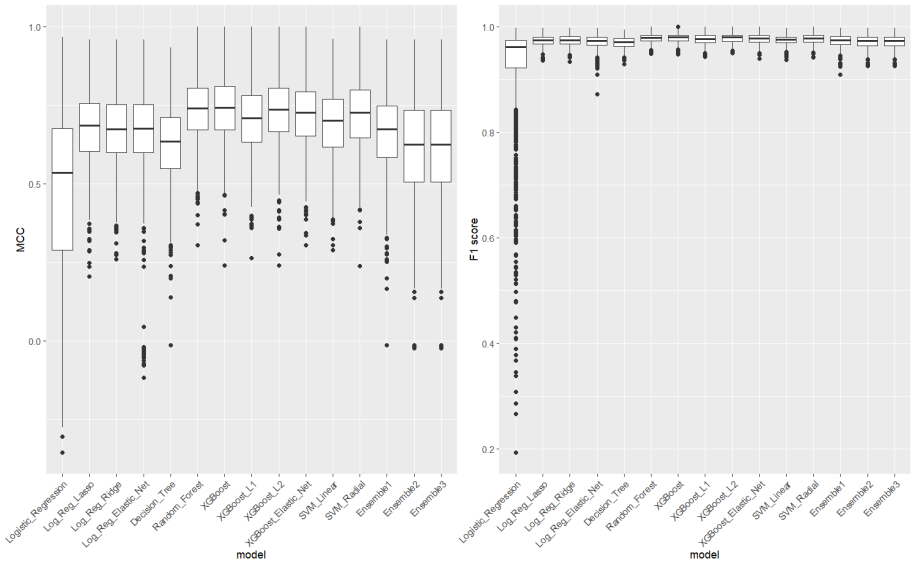


Figure 3.6: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables.

prone to outliers. In particular, the standard logistic regression model exhibited the most outliers in all scenarios. This may be due to its limited flexibility, which made it less capable of capturing complex patterns in diverse observations. Logistic regression also showed notably lower average accuracies and sensitivities, particularly in specific outlying repetitions. Moreover, although all methods exhibited relatively large IQRs for specificity, two of the ensemble models considered, along with logistic regression, exhibited substantially larger IQRs than those of any other method. A similar pattern was also evident for MCC.

When comparing different scenarios, the best prediction accuracies and F1 scores were obtained when considering *weight loss* alone or both *weight loss* and *time passed* as auxiliary variables. This matches our expectations, since *weight loss* was the variable most correlated with *TB status*. However, in terms of MCC values, using the presence of *night sweats* as the auxiliary variable produced the best results. However, the choice of auxiliary variables did not influence the results as much as the effect of data imbalance.

The boxplots of the MCC, F1 score and balanced accuracy values obtained from the neural networks described in the previous chapter are presented in Figures 3.7-3.12. Appendix C presents the corresponding boxplots of accuracy, sensitivity, and specificity. Additionally, tables that display the average values of all metrics for each neural network are included in Appendix B.

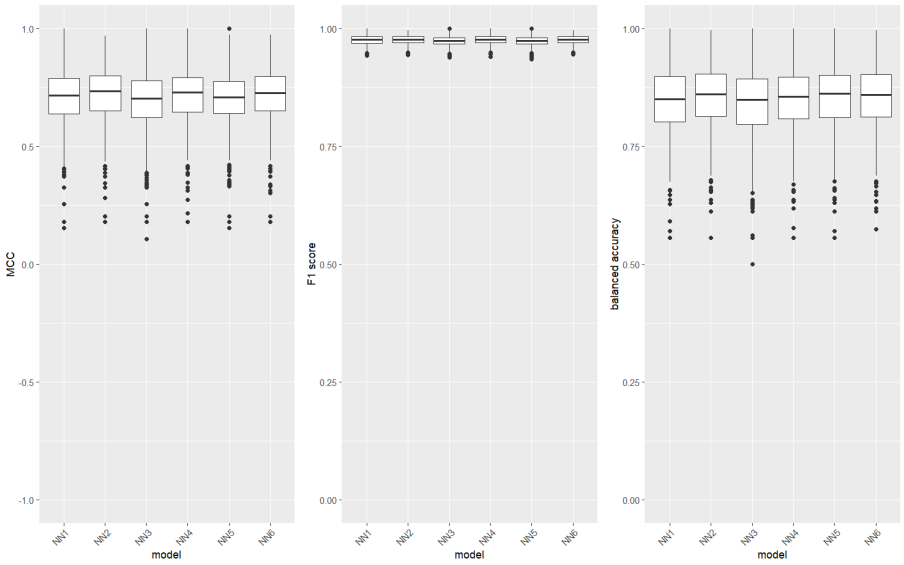


Figure 3.7: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable.

The neural networks we applied also showed relatively consistent performance, with the deeper, wider architectures and the one using LeakyReLU outperforming the other three. However, the ranking among these top-performing networks varied in different scenarios.

Nonetheless, in most of the scenarios, the MCC values of the neural networks were lower than those obtained from the best-performing ML methods, with the highest MCC value reaching only 0.770. Considering the higher complexity and considerably longer training time, XGBoost and SVM with a radial kernel might be better alternatives.

The boxplots indicate that the IQRs of the metrics across all the methods are consistent with the previous results, with F1 scores having the precise IQRs and MCC showing moderate variability. No clear patterns in the outliers were observed either, as the boxplots for different models appeared relatively similar.

Additionally, we excluded the auxiliary variables from the training dataset to assess how much their presence influenced prediction quality. The corresponding results are presented in Appendix B (Tables B.13-B.24) and Appendix C (Figures C.19-C.48). The top-performing models remained the same, but the calculated metrics were lower. Specifically, the average MCC values of XGBoost ranged from 0.368 to 0.789, random forest from

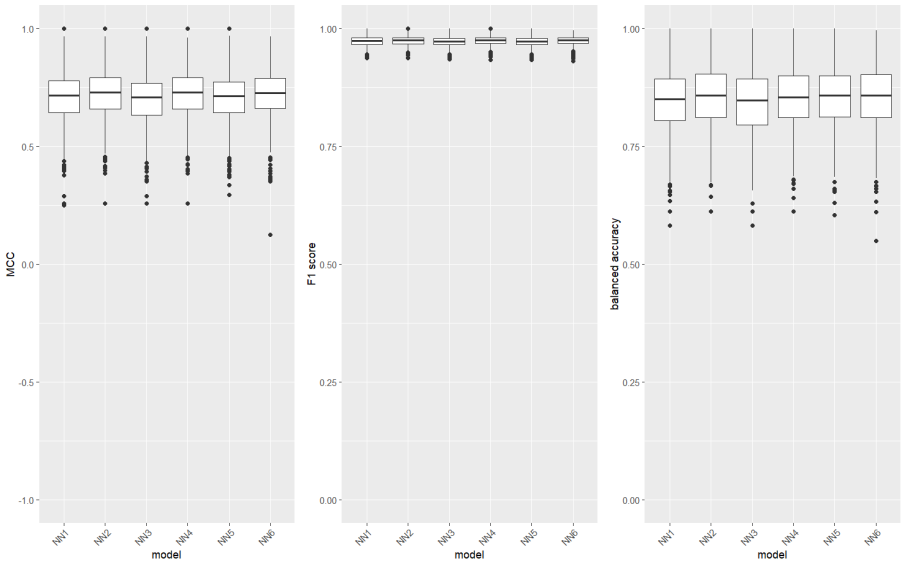


Figure 3.8: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable.

0.294 to 0.770, and SVM with a radial kernel from 0.304 to 0.773.

3.2 Real-life Data Results

We have also applied the same ML methods to the original dataset, both with and without the proposed bias correction method. Tables 3.1 and 3.2 present the prediction metrics of the methods without any bias correction. In contrast, Tables 3.3 and 3.4 show the sample results when *fever* is used as an auxiliary variable and the suggested bias correction is applied. Similarly, tables 3.5 and 3.6 show the sample results when *weight loss* is used as an auxiliary variable and the suggested bias correction is applied. Additional tables showing results for other auxiliary variables are provided in Appendix D.

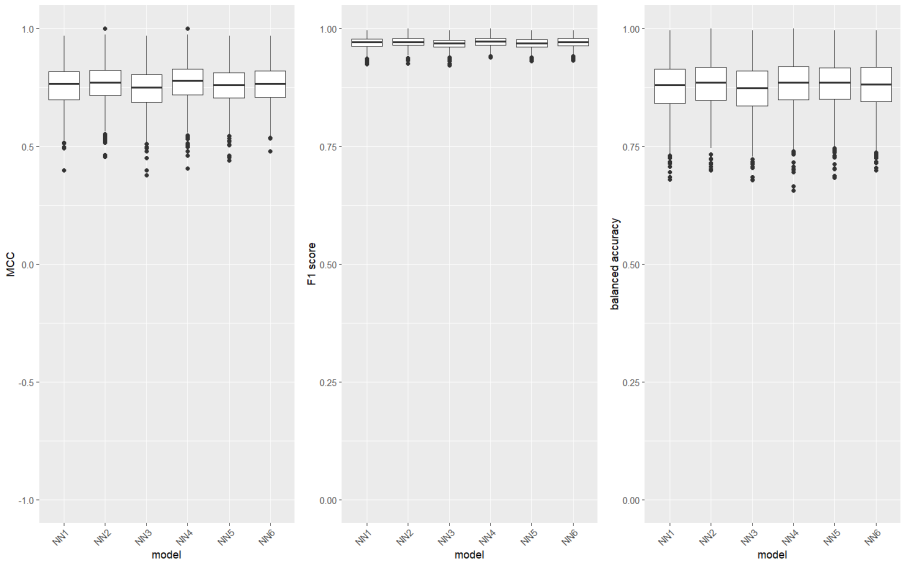


Figure 3.9: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable.

Table 3.1: Results of the ML models applied to the real-life data without bias correction

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.810	0.980	0.865	0.963	0.993	0.737
Log Reg Lasso	0.703	0.970	0.786	0.945	0.993	0.579
Log Reg Ridge	0.776	0.976	0.839	0.957	0.993	0.684
Log Reg Elastic Net	0.740	0.973	0.812	0.951	0.993	0.632
Decision Tree	0.630	0.963	0.756	0.933	0.986	0.526
Random Forest	0.612	0.959	0.776	0.927	0.972	0.579
XGBoost	0.599	0.959	0.753	0.927	0.979	0.526
XGBoost L1	0.716	0.969	0.832	0.945	0.979	0.684
XGBoost L2	0.639	0.963	0.779	0.933	0.979	0.579
XGBoost Elastic Net	0.716	0.969	0.832	0.945	0.979	0.684
SVM Linear	0.639	0.963	0.779	0.933	0.979	0.579
SVM Radial	0.708	0.969	0.809	0.945	0.986	0.632
XGBoost + Log Reg Elastic Net + SVM Linear	0.703	0.970	0.786	0.945	0.993	0.579
XGBoost Ridge + Log Reg + SVM Radial	0.740	0.973	0.812	0.951	0.993	0.632
XGBoost + Log Reg + SVM Radial	0.703	0.970	0.786	0.945	0.993	0.579

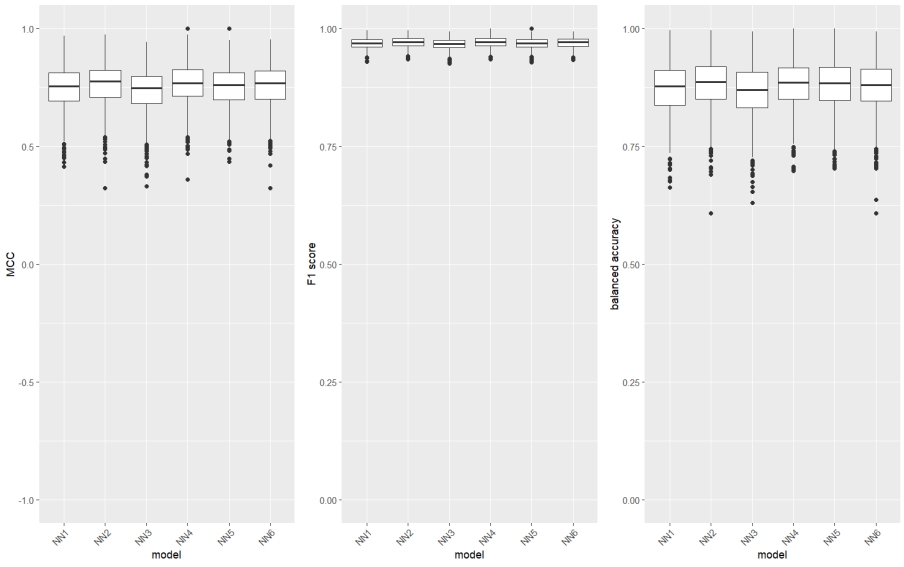


Figure 3.10: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable.

Table 3.2: Results of the neural networks applied to the real-life data without bias correction

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.703	0.970	0.786	0.945	0.993	0.579
Deeper NN (more hidden layers)	0.651	0.962	0.802	0.933	0.972	0.632
Smaller NN	0.670	0.966	0.783	0.939	0.986	0.579
Wider NN (more neurons per layer)	0.571	0.956	0.749	0.921	0.972	0.526
NN with batch normalization	0.435	0.942	0.690	0.896	0.959	0.421
NN with LeakyReLU	0.545	0.957	0.704	0.921	0.986	0.421

When no bias correction was applied, the resulting prediction MCC values of the best-performing methods were even lower than those observed in the simulations. In contrast, prediction sensitivities increased and most values reached approximately 99%. From a practical point of view, this means that as models become more sensitive to detecting positive cases, they may also produce more false positives, due to a trade-off with specificity. Therefore, if higher specificity or a better balance between sensitivity and specificity is

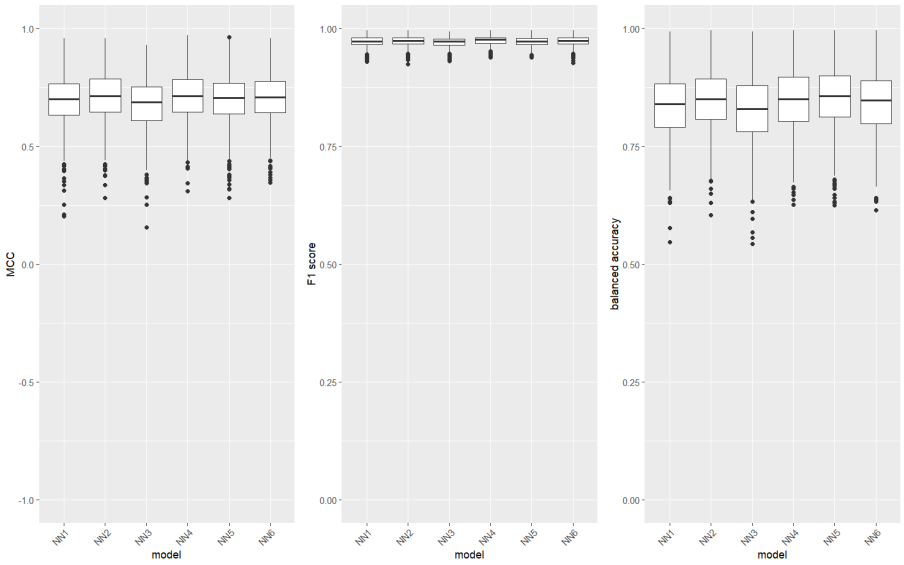


Figure 3.11: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

the primary objective, applying bias correction is advisable. Interestingly, while XGBoost, random forest, and SVM with a radial kernel were previously identified as the most consistently high-performing methods, other algorithms without bias correction outperformed them. Specifically, Logistic Regression (both in its standard form and with Ridge or elastic net regularization) and the second ensemble model achieved the highest accuracies. As for the neural networks, none achieved a higher MCC than the SVM with radial kernel, and especially not higher than the top-performing method, logistic regression.

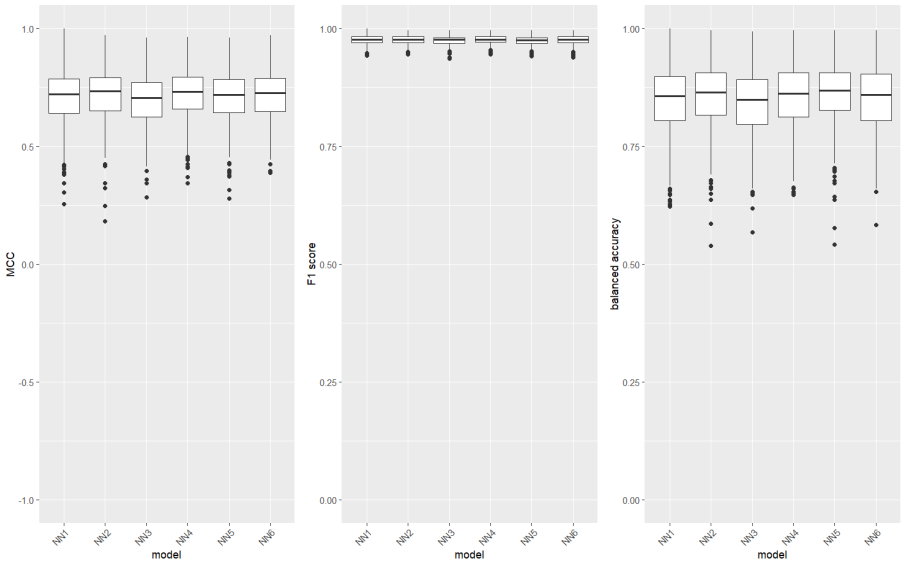


Figure 3.12: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

Table 3.5: Results from the ML models on the real-life data with *weight loss* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.652	0.981	0.862	0.963	0.974	0.750
Log Reg Lasso	0.710	0.987	0.809	0.976	0.994	0.625
Log Reg Ridge	0.515	0.981	0.684	0.963	0.994	0.375
Log Reg Elastic Net	0.710	0.987	0.809	0.976	0.994	0.625
Decision Tree	0.559	0.981	0.744	0.963	0.987	0.500
Random Forest	0.559	0.981	0.744	0.963	0.987	0.500
XGBoost	0.652	0.984	0.806	0.970	0.987	0.625
XGBoost L1	0.559	0.981	0.744	0.963	0.987	0.500
XGBoost L2	0.652	0.984	0.806	0.970	0.987	0.625
XGBoost Elastic Net	0.652	0.984	0.806	0.970	0.987	0.625
SVM Linear	0.783	0.990	0.812	0.982	1.000	0.625
SVM Radial	0.783	0.990	0.812	0.982	1.000	0.625
XGBoost + Log Reg Elastic Net + SVM Linear	0.710	0.987	0.809	0.976	0.994	0.625
XGBoost Ridge + Log Reg + SVM Radial	0.652	0.984	0.806	0.970	0.987	0.625
XGBoost + Log Reg + SVM Radial	0.652	0.984	0.806	0.970	0.987	0.625

Table 3.3: Results of the ML models applied to the real-life data with *fever* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.512	0.978	0.773	0.957	0.975	0.571
Log Reg Lasso	0.489	0.981	0.708	0.963	0.987	0.429
Log Reg Ridge	0.489	0.981	0.708	0.963	0.987	0.429
Log Reg Elastic Net	0.489	0.981	0.708	0.963	0.987	0.429
Decision Tree	0.441	0.978	0.705	0.957	0.981	0.429
Random Forest	0.611	0.981	0.844	0.963	0.975	0.714
XGBoost	0.652	0.984	0.848	0.970	0.981	0.714
XGBoost L1	0.611	0.981	0.844	0.963	0.975	0.714
XGBoost L2	0.652	0.984	0.848	0.970	0.981	0.714
XGBoost Elastic Net	0.552	0.981	0.776	0.963	0.981	0.571
SVM Linear	0.489	0.981	0.708	0.963	0.987	0.429
SVM Radial	0.358	0.978	0.636	0.957	0.987	0.286
XGBoost + Log Reg Elastic Net + SVM Linear	0.489	0.981	0.708	0.963	0.987	0.429
XGBoost Ridge + Log Reg + SVM Radial	0.441	0.978	0.705	0.957	0.981	0.429
XGBoost + Log Reg + SVM Radial	0.441	0.978	0.705	0.957	0.981	0.429

Table 3.6: Results from the neural networks on the real-life data with *weight loss* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.642	0.970	0.971	0.945	0.943	1.000
Deeper NN (more hidden layers)	0.621	0.967	0.968	0.939	0.936	1.000
Smaller NN	0.541	0.964	0.897	0.933	0.936	0.857
Wider NN (more neurons per layer)	0.621	0.967	0.968	0.939	0.936	1.000
NN with batch normalization	0.642	0.970	0.971	0.945	0.943	1.000
NN with LeakyReLU	0.541	0.964	0.897	0.933	0.936	0.857

The results of a single replication showed that although XGBoost and SVM with a radial kernel, previously identified as the best-performing methods based on the simulation, still rank among the top performers, there was some variability in the results. For example, when considering *fever* as the auxiliary variable, Table 3.3 showed that while XGBoost and random forest still achieved the highest MCC, the SVM with radial kernel performed poorly and had the lowest MCC of all methods. In contrast, when using *weight loss* as an auxiliary variable, as shown in Table 3.5, SVM with a linear or radial kernel produced the best results, and logistic regression with Lasso or elastic net regularization outperformed both random forest and XGBoost. The results of the neural networks also appear inconsistent, as Table

Table 3.4: Results of the neural networks applied to the real-life data with *fever* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.539	0.967	0.849	0.939	0.949	0.750
Deeper NN (more hidden layers)	0.593	0.967	0.909	0.939	0.942	0.875
Smaller NN	0.539	0.967	0.849	0.939	0.949	0.750
Wider NN (more neurons per layer)	0.714	0.977	0.978	0.957	0.955	1.000
NN with batch normalization	0.689	0.974	0.974	0.951	0.949	1.000
NN with LeakyReLU	0.517	0.964	0.846	0.933	0.942	0.750

3.6 indicates that the base model was among the two with the highest MCC. Interestingly, while we observed that all the methods struggled to achieve the high average specificity, some of the neural networks were able to achieve a specificity of 1 for a single repetition.

Table 3.7: Training and evaluation time of different models (in seconds)

Model	Runtime (Seconds)
Logistic Regression	0.19
Log Reg Lasso	1.17
Log Reg Ridge	0.87
Log Reg Elastic Net	1.23
Decision Tree	0.09
Random Forest	0.74
XGBoost	0.67
XGBoost L1	2.13
XGBoost L2	3.09
XGBoost Elastic Net	195.3
SVM Linear	4.44
SVM Radial	18.37
Base NN	21.80
Deeper NN (more hidden layers)	16.23
Smaller NN	12.40
Wider NN (more neurons per layer)	16.80
NN with batch normalization	17.20
NN with LeakyReLU	18.66

Table 3.7 shows the time required to train and evaluate different models. All models were run on a laptop with an Intel Core i5-8265U CPU (4 cores, 8 threads) and 8 GB of RAM, running Windows 11 and R version 4.3.2.

Overall, neural networks generally required considerably longer training times than

simpler machine learning models. The only exception is XGBoost with elastic net regularization, which took the longest. This is due to the large number of hyperparameter combinations evaluated during cross-validation to select the best combination of L1 and L2 regularization. Reducing the number of hyperparameter combinations would significantly decrease its runtime.

In general, the observations suggest that training multiple methods with bias correction is recommended to select the best model for predicting TB diagnosis. However, the decision tree model did not perform well in any scenario and can therefore be excluded from further consideration.

Finally, we performed a feature importance analysis using the same real-life data. Given that the most important features may vary across replications in the simulations, we restricted this analysis to a single replication. To simplify the results, we report only the three most important variables for the best-performing algorithm (based on MCC) in each scenario, as shown in Table D.9. Features such as *marital_status_Widowed* represent one-hot encoded variables, indicating the presence (1) or absence (0) of the corresponding category within the original categorical variable. For example, the column *marital_status_Widowed* indicates whether a person's marital status is 'widowed.' The variables *gender_1* and *gender_2* correspond to encoded gender categories, where 1 represents male and 2 represents female. Overall, we can see that *gender*, *marital status*, *time passed*, and *weight loss* were among the most important features for predicting TB.

Chapter 4

Discussions and Conclusions

4.1 Discussions

In this thesis, we explored ML-based predictive models for diagnosing TB and proposed a bias correction technique to deal with non-probability survey sampling. Many surveys use quota sampling, which needs to be addressed before proceeding to further methodology. After addressing potential bias with various auxiliary variables, we applied several classification methods to diagnose TB using the survey data and identified the models that performed the best. In addition, the use of neural networks was explored for this task. Lastly, we applied the proposed bias correction method and ML approaches to real-life non-probability survey data.

We have found that among the factors included, *weight loss* is the most strongly correlated with TB (Figure 2.1), a finding supported by several previous studies that have identified *weight loss* as one of the most common symptoms of abdominal tuberculosis [52, 53]. Davies and Pai found that, among a wider set of symptoms, only *weight loss* and *night sweats* were statistically significantly more common in the TB group compared to the non-TB group [54]. Furthermore, it has been found that there is a relationship between *weight loss* or *weight gain* and the risk of relapse of TB [55]. As a result, since *weight loss* is one of the leading indicators of TB, resampling based on the *weight loss* indicator yielded the best predictive results in terms of accuracy. However, using *night sweats* as an auxiliary variable produced the best average MCC values. Because the data are imbalanced and have a high proportion of true positives, the F1 score and some other metrics are not very informative in this case. Therefore, we prioritize MCC values when comparing the results.

The performance of the classification methods was consistent across different scenarios, with XGBoost, random forest, and the SVM with a radial kernel producing the best metrics. However, since both XGBoost and random forest are tree-based models, and XGBoost consistently performed better, it may not be beneficial to include random forest among the top-performing methods. While accuracies and sensitivities were relatively consistent across different resampled datasets, specificities varied a lot. This might be explained by the imbalance presented in the data, since considerably less than half of the participants were diagnosed with TB. The consistently low specificities reflect a high rate of false positives, which can have significant consequences in low-resource healthcare settings. Unnecessary follow-up procedures and treatments place an additional burden on already limited systems. Moreover, frequent incorrect classifications may reduce confidence in diagnostic tools and weaken public trust in the healthcare system. MCC, which takes into account all four elements of the confusion matrix, also resulted in boxplots that show considerable variability.

Interestingly, we noticed that generally, the ensembles of the models did not perform better than the top-performing single models. This could be due to class imbalance, as XGBoost handles the rare healthy cases better than the others. Differences in how the models make predictions and their varying strengths might have caused the ensemble to be less effective. These factors likely explain why ensembles did not always improve performance here.

Regarding the neural networks, although three models consistently performed better than the others, in most scenarios, their MCC values were lower than those obtained from XGBoost, random forest, and SVM with a radial kernel. Overall, all the metrics exhibited significant variability, indicated by the larger IQRs. Therefore, their higher variability makes them less reliable when relying on a single training sample. Moreover, training neural networks requires more time compared to other methods. Given this, along with the inconsistency in performance, using XGBoost or the SVM with a radial kernel may be more practical for prediction. However, this might be caused by insufficient training data, which can make even small networks struggle to train effectively. In future studies, it would be beneficial to explore how the same methodology applies to balanced datasets, as well as to data with large sample sizes.

Additionally, we observed that excluding the auxiliary variable from the training dataset did not change the ranking of the methods but lowered the metric values.

We also found that variables such as *gender*, *marital status*, *time passed* since the participant began experiencing symptoms, and the presence of *weight loss* were among the most important. It is surprising to see *marital status* as one of the key factors, and this

may be due to several reasons. First, this variable was systematically missing, as it was only missing for participants from India. Additionally, there may be some confounding factors (for example, household size could be related to marital status [56]).

As mentioned in Section 1.3, Buelens et al. applied a similar bias correction approach in the context of a regression problem with a continuous outcome variable [34]. In their study, results were evaluated against known population-level values, which are unavailable in our case. Nonetheless, our work extended their approach by also addressing model selection and the choice of auxiliary variables in the context of predicting binary outcome variables. Additionally, the highest accuracies and F1 scores obtained in our simulations exceed those reported in a similar study by Karmani et al. [22].

Our study demonstrates promising results in diagnosing TB with computational methods, even when limited participant characteristics are available. Furthermore, we showed that the bias correction method described in Section 2.3.1 is effective when applied to categorical variables. This approach may be particularly valuable in analyses based on questionnaire data collected through quota sampling – a common practice in real-world settings.

One of the key strengths of this study is that, to our knowledge, it is the first to adapt the described bias correction technique to a binary outcome setting. In addition, this work is the first to apply this approach in the context of TB prediction, extending its applicability to a significant real-world public health problem. Despite its contributions, this study also has limitations. First, the relatively small sample size and the limited number of healthy individuals may have impacted the ability of the predictive models to generalize, contributing to the reduced specificity. Second, given the age calculation process described in Section 2.2, inaccuracies may occur when using *age* as a variable for prediction. Also, while we identified methods that performed best on average, the process of applying bias correction to real-world data remains somewhat exploratory, relying on iterative testing of different approaches rather than a standardized protocol. Additionally, the population data we considered is biased, as it was collected using quota sampling. Finally, although some previous studies on ML and computer vision-based TB diagnosis reported superior predictive performance, these results were obtained in different contexts and often relied on probability-based sampling, limiting their direct comparability with our findings.

Future research could address these limitations in several ways. In particular, exploring alternative bias correction methodologies and adapting them to binary outcomes could lead to improved performance and greater robustness across different data sets. Future studies could also explore applying methods to address class imbalance, such as the Synthetic

Minority Oversampling Technique (SMOTE) or cost-sensitive learning approaches that modify the learning algorithm to penalize misclassifications of the minority class more heavily. Next, utilizing the multiple imputation technique to handle missing values may be beneficial. Developing more systematic procedures for model selection and bias adjustment in real-world settings would also enhance reproducibility and reduce reliance on trial-and-error. Additionally, as more comprehensive and balanced datasets become available, they may enable more robust model training and validation.

4.2 Conclusions

In summary, this thesis highlights the importance of incorporating bias correction when applying ML methods to diagnosis tasks using non-randomly sampled data. Our findings showed that failing to address sampling bias can lead to misleading model performance. The adapted correction technique for binary outcomes demonstrated substantial potential, especially in the context of TB screening based on limited survey information. These results underscore the need for future studies not only to consider bias correction as a necessary step in real-world analyses, but also to refine and expand the available methodologies to achieve greater impact.

References

[1] World Health Organization. Global tuberculosis report 2024, 2024.

[2] Elisa Farina, Carmen D'Amore, Laura Lancella, Elena Boccuzzi, Marta Luisa Ciofi degli Atti, Antonino Reale, Paolo Rossi, Alberto Villani, Massimiliano Raponi, and Umberto Raucci. Alert sign and symptoms for the early diagnosis of pulmonary tuberculosis: analysis of patients followed by a tertiary pediatric hospital. *Italian Journal of Pediatrics*, 48(1):90, 2022.

[3] Cesar Ugarte-Gil, Rodrigo M Carrillo-Larco, and Daniela E Kirwan. Latent tuberculosis infection and non-infectious co-morbidities: Diabetes mellitus type 2, chronic kidney disease and rheumatoid arthritis. *International Journal of Infectious Diseases*, 80:S29–S31, 2019.

[4] Anne O'Garra, Paul S Redford, Finlay W McNab, Chloe I Bloom, Robert J Wilkinson, and Matthew PR Berry. The immune response in tuberculosis. *Annual review of immunology*, 31(1):475–527, 2013.

[5] Ole Magnus Bjørgaas Helle, Mala Kanthali, Erlend Grønningen, Shoaib Hassan, Manju Raj Purohit, and Tehmina Mustafa. Factors associated with hospitalization and mortality in adult and pediatric extrapulmonary tuberculosis at a tertiary care hospital in central india. *Infectious Diseases*, 56(12):1080–1092, 2024.

[6] World Health Organization. *WHO consolidated guidelines on tuberculosis. Module 2: screening-systematic screening for tuberculosis disease*. World Health Organization, 2021.

[7] Jessy Lallungawi Khawbung, Durbba Nath, and Supriyo Chakraborty. Drug resistant tuberculosis: A review. *Comparative immunology, microbiology and infectious diseases*, 74:101574, 2021.

[8] Edine W Tiemersma, Marieke J van der Werf, Martien W Borgdorff, Brian G Williams, and Nico JD Nagelkerke. Natural history of tuberculosis: duration and fatality of untreated pulmonary tuberculosis in hiv negative patients: a systematic review. *PloS one*, 6(4):e17601, 2011.

[9] World Health Organization. Implementing the end tb strategy: The essentials 2022, 2022.

[10] World Health Organization. *WHO consolidated guidelines on tuberculosis. Module 3: diagnosis*. World Health Organization, 2025.

[11] Anil M Solanki, Saumyajit Basu, Amitava Biswas, Subhendu Roy, and Aditya Banta. Sensitivity and specificity of gene xpert in the diagnosis of spinal tuberculosis: a prospective controlled clinical study. *Global Spine Journal*, 10(5):553–558, 2020.

[12] Anastasia A Theodosiou and Robert C Read. Artificial intelligence, machine learning and deep learning: Potential resources for the infection clinician. *Journal of Infection*, 2023.

[13] Joan Jonathan and Alcardo Alex Barakabitze. Ml technologies for diagnosing and treatment of tuberculosis: a survey. *Health and Technology*, 13(1):17–33, 2023.

[14] Manisha Singh, Gurubasavaraj Veeranna Pujar, Sethu Arun Kumar, Meduri Bhagyalalitha, Handattu Shankaranarayana Akshatha, Belal Abuhaija, Anas Ratib Alsoud, Laith Abualigah, Narasimha M Beeraka, and Amir H Gandomi. Evolution of machine learning in tuberculosis diagnosis: a review of deep learning-based medical applications. *Electronics*, 11(17):2634, 2022.

[15] Syeda Shaizadi Meraj, Razali Yaakob, Azreen Azman, Siti Nuralain Mohd Rum, and AS Ahmad Nazri. Artificial intelligence in diagnosing tuberculosis: a review. *International Journal on Advanced Science, Engineering and Information Technology*, 9(1):81–91, 2019.

[16] Ali A El-Solh, Chiu-Bin Hsiao, Susan Goodnough, Joseph Serghani, and Brydon JB Grant. Predicting active pulmonary tuberculosis using an artificial neural network. *Chest*, 116(4):968–973, 1999.

[17] Tanvir Mahtab Zihan, Abduz Zami, Mohiuddin Ahmed, Rakibul Islam, and Fahim Ahmed. From pixels to diagnosis: Convolutional neural networks in tuberculosis screening. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pages 1187–1192. IEEE, 2024.

[18] Angela Houston and Derek Clive Macallan. Extrapulmonary tuberculosis. *Medicine*, 42(1):18–22, 2014.

[19] Robert L Serafino Wani. Clinical manifestations of pulmonary and extra-pulmonary tuberculosis. *South Sudan Medical Journal*, 6(3):52–56, 2013.

[20] Erhan Elveren and Nejat Yumuşak. Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm. *Journal of medical systems*, 35:329–332, 2011.

[21] Maicon Herverton Lino Ferreira da Silva Barros, Geovanne Oliveira Alves, Lubnnia Morais Florêncio Souza, Elisson da Silva Rocha, João Fausto Lorenzato de Oliveira, Theo Lynn, Vanderson Sampaio, and Patricia Takako Endo. Benchmarking machine learning models to assist in the prognosis of tuberculosis. In *Informatics*, volume 8, page 27. MDPI, 2021.

[22] Priyanka Karmani, Aftab Ahmed Chandio, Imtiaz Ali Korejo, Oluwarotimi Williams Samuel, and Majed Aborokbah. Machine learning based tuberculosis (ml-tb) health predictor model: early tb health disease prediction with ml models for prevention in developing countries. *PeerJ Computer Science*, 10:e2397, 2024.

[23] Ali Akbar Septiandri, Aditiawarman, Roy Tjiong, Erlina Burhan, and Anuraj H Shankar. Cost-sensitive machine learning classification for mass tuberculosis screening. *medRxiv*, page 19000190, 2019.

[24] Victor Hugo Gutierrez-Velez and Daniel Wiese. Sampling bias mitigation for species occurrence modeling using machine learning methods. *Ecological Informatics*, 58:101091, 2020.

[25] Wendi Liu, Svetlana Ikonnikova, H Scott Hamlin, Livia Sivila, and Michael J Pyrcz. Demonstration and mitigation of spatial sampling bias for machine-learning predictions. *SPE Reservoir Evaluation & Engineering*, 24(01):262–274, 2021.

[26] Roshni Sahoo, Lihua Lei, and Stefan Wager. Learning from a biased sample. *arXiv preprint arXiv:2209.01754*, 2022.

[27] Ali Rafei, Michael R Elliott, and Carol AC Flannagan. Robust model-based inference for non-probability samples. *arXiv preprint arXiv:2204.03215*, 2022.

[28] Yutao Liu, Andrew Gelman, and Qixuan Chen. Inference from nonrandom samples using bayesian machine learning. *Journal of Survey Statistics and Methodology*, 11(2):433–455, 2023.

[29] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[30] Surya Mattu Julia Angwin, Jeff Larson and Lauren Kirchner. Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016. ProPublica.

[31] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 35–50. Springer, 2012.

[32] Yongkai Wu, Lu Zhang, and Xintao Wu. Fairness-aware classification: Criterion, convexity, and bounds. *arXiv preprint arXiv:1809.04737*, 2018.

[33] Ninareh Mehrabi, Umang Gupta, Fred Morstatter, Greg Ver Steeg, and Aram Galstyan. Attributing fair decisions with attention interventions. *arXiv preprint arXiv:2109.03952*, 2021.

[34] Bart Buelens, JMS Burger, and Jan A van den Brakel. *Predictive inference for non-probability samples: a simulation study*, volume 13. Statistics Netherlands The Hague, 2015.

[35] Lavanya Huria, Bony Wiem Lestari, Eka Saptiningrum, Auliya Ramanda Fikri, Charity Oga-Omenka, Mohammad Abdullah Heel Kafi, Benjamin Daniels, Nathaly Aguilera Vasquez, Angelina Sassi, Jishnu Das, et al. Care pathways of individuals with tuberculosis before and during the covid-19 pandemic in bandung, indonesia. *PLOS Global Public Health*, 4(1):e0002251, 2024.

[36] Charity Oga-Omenka, Lauren Rosapep, Elaine Baruwa, Lavanya Huria, Nathaly Aquilera Vasquez, Bolanle Olusola Faleye, Md Abdullah Heel Kafi, Angelina Sassi, Chimdi Nwosu, Benjamin Johns, et al. Individual journeys to tuberculosis care in nigeria's private sector during the covid-19 pandemic. *BMJ Global Health*, 9(1):e013124, 2024.

[37] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232, 1958.

[38] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[39] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[40] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

[41] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[42] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[43] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[44] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[45] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

[46] Bradley Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association*, 78(382):316–331, 1983.

[47] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems*, 26, 2013.

[48] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[49] Trevor Hastie. The elements of statistical learning: data mining, inference, and prediction, 2009.

[50] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[51] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, 2020.

[52] Ali Uzunkoy, Muge Harma, and Mehmet Harma. Diagnosis of abdominal tuberculosis: experience from 11 cases and review of the literature. *World journal of gastroenterology: WJG*, 10(24):3647, 2004.

[53] S Rai and WM Thomas. Diagnosis of abdominal tuberculosis: the importance of laparoscopy. *Journal of the Royal Society of Medicine*, 96(12):586–588, 2003.

[54] PDO Davies and M Pai. The diagnosis and misdiagnosis of tuberculosis [state of the art series. tuberculosis. edited by id rusen. number 1 in the series]. *The International Journal of Tuberculosis and Lung Disease*, 12(11):1226–1234, 2008.

[55] Awal Khan, Timothy R Sterling, Randall Reves, Andrew Vernon, C Robert Horsburgh, and Tuberculosis Trials Consortium. Lack of weight gain and relapse risk in a large tuberculosis treatment trial. *American journal of respiratory and critical care medicine*, 174(3):344–348, 2006.

[56] Sean M Viña. Diminished psychedelic returns on distress: Marital status and household size. *PLoS One*, 19(3):e0293675, 2024.

[57] H Simon Schaaf, Andrea Collins, Adrie Bekker, and Peter DO Davies. Tuberculosis at extremes of age. *Respirology*, 15(5):747–763, 2010.

APPENDICES

Appendix A

Neural Network Models Architectures for Section 2.3.3.

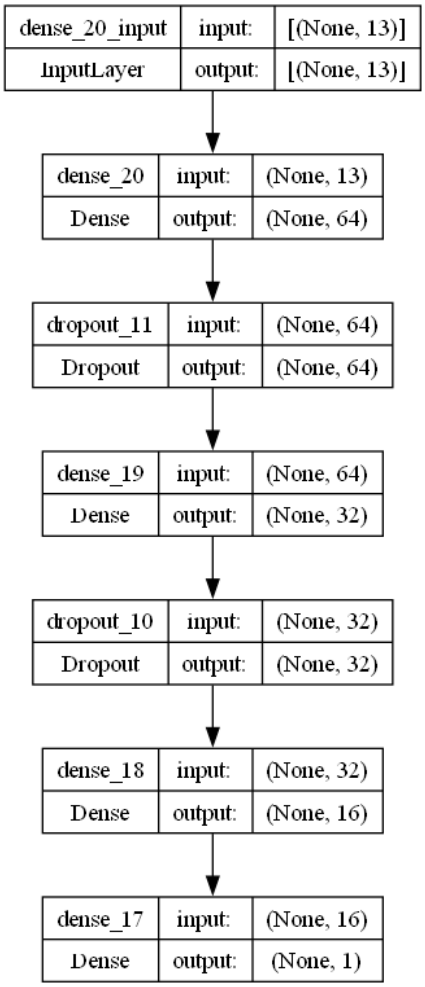


Figure A.1: The architecture of the second (deeper) model.

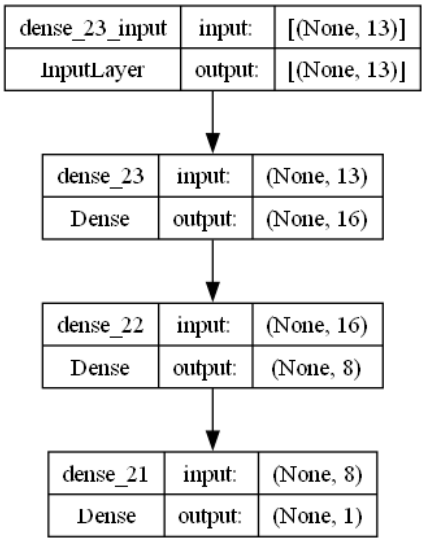


Figure A.2: The architecture of the third (smaller) model.

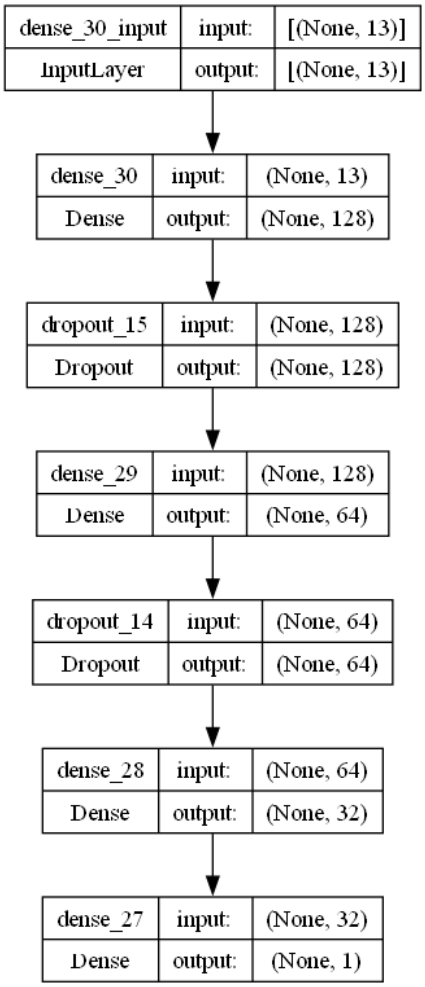


Figure A.3: The architecture of the fourth (wider) model.

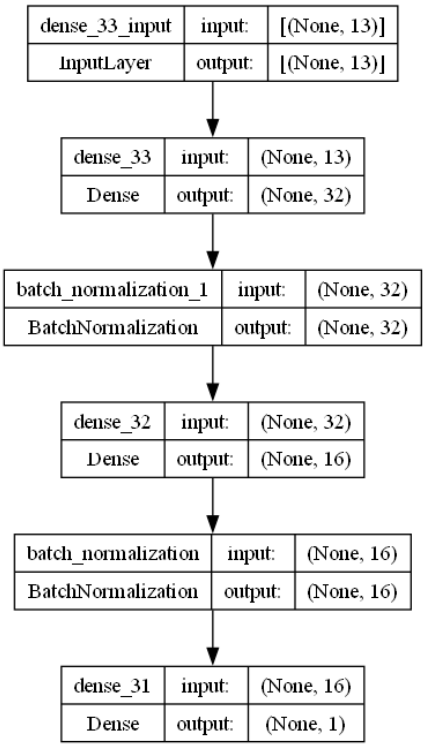


Figure A.4: The architecture of the fifth model (with batch normalization).

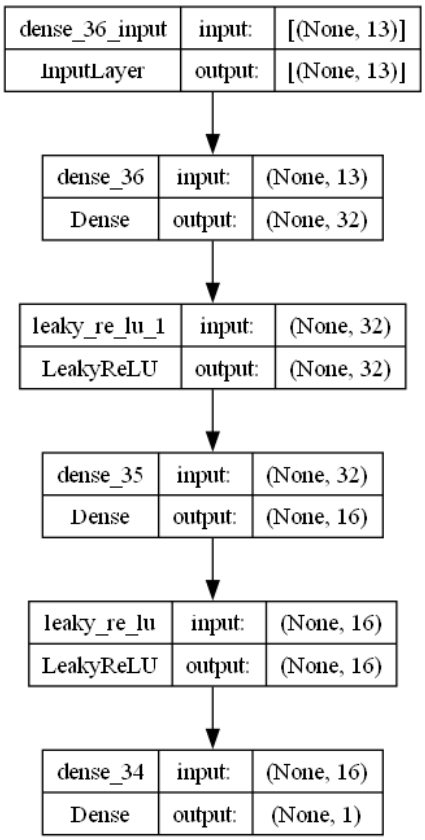


Figure A.5: The architecture of the fifth model (with LeakyReLU).

Appendix B

Additional Simulation Results

Table B.1: Average results when considering *weight loss* as an auxiliary variable (oversample probability=0.61, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.488	0.919	0.746	0.874	0.901	0.591
Log Reg Lasso	0.674	0.973	0.797	0.896	0.917	0.678
Log Reg Ridge	0.659	0.973	0.783	0.897	0.921	0.645
Log Reg Elastic Net	0.670	0.973	0.794	0.903	0.926	0.663
Decision Tree	0.626	0.969	0.793	0.944	0.976	0.610
Random Forest	**0.728**	**0.977**	**0.856**	**0.958**	**0.980**	**0.733**
XGBoost	**0.740**	**0.978**	**0.865**	**0.960**	**0.980**	**0.751**
XGBoost L1	0.700	0.975	0.836	0.954	0.980	0.692
XGBoost L2	0.738	0.978	0.863	0.959	0.980	0.746
XGBoost Elastic Net	0.724	0.977	0.853	0.957	0.980	0.727
SVM Linear	0.693	0.973	0.845	0.952	0.975	0.714
SVM Radial	**0.723**	**0.977**	**0.852**	**0.957**	**0.980**	**0.724**
XGBoost + Log Reg Elastic Net + SVM Linear	0.664	0.974	0.779	0.952	0.989	0.570
XGBoost Ridge + Log Reg + SVM Radial	0.612	0.971	0.735	0.946	0.991	0.478
XGBoost + Log Reg + SVM Radial	0.612	0.971	0.735	0.946	0.991	0.478

Table B.2: Average results from neural networks when considering *weight loss* as an auxiliary variable (oversample probability=0.61, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.708	0.975	0.848	0.955	0.977	0.719
Deeper NN (more hidden layers)	**0.722**	**0.976**	**0.856**	**0.957**	**0.978**	**0.735**
Smaller NN	0.696	0.974	0.842	0.953	0.976	0.708
Wider NN (more neurons per layer)	0.715	0.976	0.851	0.956	0.978	0.725
NN with batch normalization	0.702	0.974	0.855	0.952	0.973	0.737
NN with LeakyReLU	**0.718**	**0.976**	**0.854**	**0.956**	**0.977**	**0.731**

Table B.3: Average results when considering *time passed* as an auxiliary variable (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.469	0.910	0.736	0.862	0.890	0.582
Log Reg Lasso	0.624	0.969	0.766	0.880	0.906	0.627
Log Reg Ridge	0.609	0.968	0.760	0.905	0.938	0.582
Log Reg Elastic Net	0.618	0.968	0.768	0.903	0.932	0.604
Decision Tree	0.576	0.964	0.769	0.934	0.971	0.568
Random Forest	**0.711**	**0.975**	**0.845**	**0.954**	**0.978**	**0.711**
XGBoost	**0.731**	**0.976**	**0.858**	**0.956**	**0.979**	**0.738**
XGBoost L1	0.672	0.972	0.821	0.949	0.977	0.664
XGBoost L2	0.725	0.976	0.854	0.956	0.979	0.729
XGBoost Elastic Net	0.711	0.975	0.846	0.954	0.978	0.713
SVM Linear	0.659	0.970	0.824	0.945	0.972	0.676
SVM Radial	**0.711**	**0.974**	**0.846**	**0.954**	**0.978**	**0.714**
XGBoost + Log Reg Elastic Net + SVM Linear	0.612	0.970	0.745	0.944	0.988	0.502
XGBoost Ridge + Log Reg + SVM Radial	0.585	0.968	0.724	0.941	0.990	0.458
XGBoost + Log Reg + SVM Radial	0.585	0.968	0.724	0.941	0.990	0.458

Table B.4: Average results from neural networks when considering *time passed* as an auxiliary variable (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.692	0.973	0.836	0.951	0.976	0.695
Deeper NN (more hidden layers)	**0.709**	**0.974**	**0.848**	**0.953**	**0.976**	**0.719**
Smaller NN	0.675	0.971	0.828	0.948	0.975	0.680
Wider NN (more neurons per layer)	**0.713**	**0.974**	**0.848**	**0.954**	**0.977**	**0.719**
NN with batch normalization	0.698	0.972	0.852	0.950	0.972	0.732
NN with LeakyReLU	0.704	0.974	0.844	0.952	0.976	0.711

Table B.5: Average results when considering *fever* as an auxiliary variable (inclusion probability=0.67, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.500	0.921	0.750	0.877	0.906	0.594
Log Reg Lasso	0.671	0.971	0.801	0.913	0.938	0.664
Log Reg Ridge	0.652	0.970	0.782	0.906	0.934	0.630
Log Reg Elastic Net	0.664	0.971	0.798	0.914	0.941	0.654
Decision Tree	0.612	0.966	0.786	0.938	0.974	0.598
Random Forest	**0.732**	**0.976**	**0.856**	**0.956**	**0.979**	**0.734**
XGBoost	**0.746**	**0.977**	**0.867**	**0.958**	**0.979**	**0.754**
XGBoost L1	0.698	0.973	0.834	0.951	0.978	0.689
XGBoost L2	0.743	0.976	0.864	0.957	0.979	0.748
XGBoost Elastic Net	0.733	0.976	0.857	0.956	0.979	0.735
SVM Linear	0.690	0.971	0.839	0.948	0.974	0.705
SVM Radial	**0.730**	**0.976**	**0.853**	**0.956**	**0.980**	**0.727**
XGBoost + Log Reg Elastic Net + SVM Linear	0.660	0.972	0.778	0.949	0.988	0.567
XGBoost Ridge + Log Reg + SVM Radial	0.614	0.969	0.737	0.943	0.991	0.483
XGBoost + Log Reg + SVM Radial	0.614	0.969	0.737	0.943	0.991	0.483

Table B.6: Average results from neural networks when considering *fever* as an auxiliary variable (oversample probability=0.67, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.708	0.973	0.846	0.952	0.976	0.715
Deeper NN (more hidden layers)	0.720	0.974	0.854	0.953	0.977	0.731
Smaller NN	0.698	0.972	0.841	0.950	0.975	0.708
Wider NN (more neurons per layer)	**0.721**	**0.974**	**0.853**	**0.954**	**0.977**	**0.730**
NN with batch normalization	0.704	0.972	0.855	0.950	0.972	0.738
NN with LeakyReLU	**0.721**	**0.974**	**0.854**	**0.954**	**0.977**	**0.731**

Table B.7: Average results when considering *night sweats* as an auxiliary variable (inclusion probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.564	0.921	0.783	0.878	0.909	0.658
Log Reg Lasso	0.717	0.966	0.848	0.936	0.965	0.730
Log Reg Ridge	0.701	0.965	0.831	0.932	0.966	0.695
Log Reg Elastic Net	0.715	0.965	0.846	0.933	0.962	0.730
Decision Tree	0.675	0.961	0.828	0.931	0.965	0.691
Random Forest	**0.777**	**0.972**	**0.888**	**0.951**	**0.973**	**0.803**
XGBoost	**0.794**	**0.975**	**0.894**	**0.955**	**0.976**	**0.811**
XGBoost L1	0.747	0.969	0.868	0.946	0.972	0.763
XGBoost L2	0.789	0.974	0.891	0.955	0.976	0.806
XGBoost Elastic Net	0.778	0.973	0.886	0.952	0.974	0.798
SVM Linear	0.731	0.966	0.867	0.941	0.966	0.768
SVM Radial	**0.771**	**0.972**	**0.884**	**0.951**	**0.973**	**0.796**
XGBoost + Log Reg Elastic Net + SVM Linear	0.718	0.968	0.820	0.943	0.985	0.656
XGBoost Ridge + Log Reg + SVM Radial	0.666	0.964	0.776	0.935	0.987	0.565
XGBoost + Log Reg + SVM Radial	0.666	0.964	0.776	0.935	0.987	0.565

Table B.8: Average results from neural networks when considering *night sweats* as an auxiliary variable (oversample probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.755	0.970	0.875	0.947	0.971	0.779
Deeper NN (more hidden layers)	**0.766**	**0.971**	**0.882**	**0.949**	**0.972**	**0.791**
Smaller NN	0.743	0.968	0.870	0.944	0.969	0.772
Wider NN (more neurons per layer)	**0.770**	**0.972**	**0.882**	**0.950**	**0.973**	**0.790**
NN with batch normalization	0.754	0.969	0.882	0.946	0.967	0.796
NN with LeakyReLU	0.763	0.971	0.880	0.949	0.972	0.789

Table B.9: Average results when considering *lack of appetite* as an auxiliary variable (inclusion probability=0.34)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.551	0.921	0.778	0.876	0.908	0.649
Log Reg Lasso	0.706	0.965	0.843	0.938	0.970	0.715
Log Reg Ridge	0.689	0.963	0.827	0.936	0.972	0.682
Log Reg Elastic Net	0.703	0.964	0.841	0.937	0.969	0.713
Decision Tree	0.675	0.961	0.828	0.931	0.966	0.691
Random Forest	**0.772**	**0.972**	**0.884**	**0.951**	**0.973**	**0.795**
XGBoost	**0.788**	**0.974**	**0.890**	**0.954**	**0.976**	**0.805**
XGBoost L1	0.745	0.969	0.867	0.946	0.972	0.761
XGBoost L2	0.784	0.974	0.888	0.954	0.975	0.801
XGBoost Elastic Net	0.776	0.972	0.884	0.952	0.974	0.794
SVM Linear	0.723	0.965	0.864	0.940	0.965	0.764
SVM Radial	**0.770**	**0.972**	**0.882**	**0.950**	**0.973**	**0.792**
XGBoost + Log Reg Elastic Net + SVM Linear	0.706	0.967	0.813	0.941	0.984	0.642
XGBoost Ridge + Log Reg + SVM Radial	0.660	0.963	0.772	0.933	0.987	0.556
XGBoost + Log Reg + SVM Radial	0.660	0.963	0.772	0.933	0.987	0.556

Table B.10: Average results from neural networks when considering *lack of appetite* as an auxiliary variable (oversample probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.749	0.969	0.874	0.946	0.970	0.777
Deeper NN (more hidden layers)	**0.764**	**0.971**	**0.881**	**0.949**	**0.971**	**0.791**
Smaller NN	0.735	0.967	0.867	0.942	0.967	0.767
Wider NN (more neurons per layer)	**0.765**	**0.971**	**0.881**	**0.949**	**0.972**	**0.789**
NN with batch normalization	0.753	0.969	0.880	0.946	0.967	0.793
NN with LeakyReLU	0.757	0.970	0.877	0.947	0.971	0.782

Table B.11: Average results when considering *weight loss and time passed* as auxiliary variables (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.470	0.912	0.742	0.864	0.889	0.595
Log Reg Lasso	0.674	0.974	0.732	0.709	0.703	0.760
Log Reg Ridge	0.666	0.974	0.752	0.799	0.808	0.696
Log Reg Elastic Net	0.656	0.971	0.736	0.806	0.821	0.651
Decision Tree	0.626	0.970	0.802	0.944	0.973	0.630
Random Forest	**0.732**	**0.977**	**0.865**	**0.959**	**0.978**	**0.753**
XGBoost	**0.734**	**0.978**	**0.864**	**0.959**	**0.979**	**0.748**
XGBoost L1	0.700	0.975	0.840	0.955	0.978	0.702
XGBoost L2	0.729	0.977	0.860	0.959	0.979	0.741
XGBoost Elastic Net	0.718	0.977	0.852	0.957	0.979	0.726
SVM Linear	0.689	0.974	0.844	0.952	0.975	0.713
SVM Radial	**0.720**	**0.977**	**0.852**	**0.958**	**0.979**	**0.725**
XGBoost + Log Reg Elastic Net + SVM Linear	0.660	0.973	0.755	0.949	0.989	0.522
XGBoost Ridge + Log Reg + SVM Radial	0.606	0.971	0.732	0.946	0.991	0.474
XGBoost + Log Reg + SVM Radial	0.606	0.971	0.732	0.946	0.991	0.474

Table B.12: Average results from neural networks when considering *weight loss and time passed* as auxiliary variables (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.709	0.976	0.850	0.956	0.978	0.722
Deeper NN (more hidden layers)	**0.723**	**0.977**	**0.860**	**0.957**	**0.978**	**0.742**
Smaller NN	0.697	0.975	0.844	0.954	0.977	0.711
Wider NN (more neurons per layer)	**0.724**	**0.977**	**0.857**	**0.958**	**0.979**	**0.736**
NN with batch normalization	0.710	0.975	0.863	0.954	0.973	0.753
NN with LeakyReLU	0.717	0.976	0.854	0.957	0.978	0.730

Table B.13: Average results when considering *weight loss* as an auxiliary variable and excluding it from the training data (oversample probability=0.61, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.446	0.922	0.716	0.876	0.910	0.522
Log Reg Lasso	0.593	0.968	0.744	0.872	0.899	0.588
Log Reg Ridge	0.575	0.968	0.727	0.876	0.907	0.546
Log Reg Elastic Net	0.589	0.968	0.742	0.885	0.916	0.568
Decision Tree	0.573	0.966	0.764	0.937	0.974	0.553
Random Forest	**0.644**	**0.972**	**0.783**	**0.949**	**0.985**	**0.580**
XGBoost	**0.685**	**0.974**	**0.834**	**0.952**	**0.977**	**0.691**
XGBoost L1	0.640	0.971	0.797	0.947	0.979	0.615
XGBoost L2	0.683	0.973	0.831	0.952	0.977	0.685
XGBoost Elastic Net	0.671	0.973	0.822	0.950	0.978	0.666
SVM Linear	0.609	0.967	0.793	0.940	0.972	0.615
SVM Radial	**0.661**	**0.972**	**0.813**	**0.949**	**0.978**	**0.648**
XGBoost + Log Reg Elastic Net + SVM Linear	0.583	0.969	0.724	0.943	0.990	0.459
XGBoost Ridge + Log Reg + SVM Radial	0.550	0.968	0.698	0.940	0.992	0.404
XGBoost + Log Reg + SVM Radial	0.550	0.968	0.698	0.940	0.992	0.404

Table B.14: Average results from neural networks when considering *weight loss* as an auxiliary variable and excluding it from the training data (oversample probability=0.61, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.657	0.972	0.811	0.948	0.978	0.645
Deeper NN (more hidden layers)	0.671	0.972	0.825	0.950	0.976	0.675
Smaller NN	0.644	0.971	0.806	0.946	0.976	0.636
Wider NN (more neurons per layer)	0.671	0.972	0.827	0.949	0.975	0.679
NN with batch normalization	0.656	0.970	0.830	0.945	0.970	0.691
NN with LeakyReLU	0.669	0.972	0.824	0.949	0.976	0.673

Table B.15: Average results when considering *time passed* as an auxiliary variable and excluding it from the training data (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.425	0.922	0.710	0.875	0.912	0.507
Log Reg Lasso	0.464	0.957	0.693	0.878	0.920	0.465
Log Reg Ridge	0.452	0.958	0.677	0.851	0.891	0.464
Log Reg Elastic Net	0.461	0.956	0.690	0.871	0.912	0.467
Decision Tree	0.510	0.957	0.748	0.922	0.961	0.534
Random Forest	**0.551**	**0.962**	**0.763**	**0.930**	**0.968**	**0.558**
XGBoost	**0.581**	**0.962**	**0.790**	**0.932**	**0.963**	**0.616**
XGBoost L1	0.537	0.960	0.757	0.928	0.966	0.547
XGBoost L2	0.578	0.962	0.787	0.932	0.964	0.610
XGBoost Elastic Net	0.557	0.961	0.772	0.930	0.965	0.579
SVM Linear	0.533	0.958	0.768	0.924	0.959	0.578
SVM Radial	**0.565**	**0.961**	**0.773**	**0.929**	**0.964**	**0.582**
XGBoost + Log Reg Elastic Net + SVM Linear	0.448	0.959	0.668	0.924	0.982	0.354
XGBoost Ridge + Log Reg + SVM Radial	0.468	0.959	0.668	0.925	0.982	0.355
XGBoost + Log Reg + SVM Radial	0.468	0.959	0.668	0.925	0.982	0.354

Table B.16: Average results from neural networks when considering *time passed* as an auxiliary variable and excluding it from the training data (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.550	0.960	0.770	0.928	0.963	0.576
Deeper NN (more hidden layers)	0.578	0.961	0.797	0.929	0.958	0.637
Smaller NN	0.536	0.960	0.760	0.927	0.964	0.555
Wider NN (more neurons per layer)	0.574	0.961	0.791	0.929	0.960	0.622
NN with batch normalization	0.573	0.960	0.798	0.927	0.956	0.639
NN with LeakyReLU	0.560	0.961	0.775	0.929	0.963	0.587

Table B.17: Average results when considering *fever* as an auxiliary variable and excluding it from the training data (inclusion probability=0.67, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.506	0.926	0.753	0.883	0.913	0.593
Log Reg Lasso	0.671	0.971	0.800	0.909	0.935	0.665
Log Reg Ridge	0.653	0.970	0.782	0.905	0.933	0.630
Log Reg Elastic Net	0.669	0.971	0.798	0.910	0.937	0.660
Decision Tree	0.617	0.966	0.787	0.939	0.974	0.601
Random Forest	**0.722**	**0.975**	**0.847**	**0.955**	**0.980**	**0.714**
XGBoost	**0.747**	**0.977**	**0.868**	**0.958**	**0.979**	**0.758**
XGBoost L1	0.701	0.973	0.836	0.951	0.978	0.694
XGBoost L2	0.746	0.977	0.867	0.958	0.979	0.756
XGBoost Elastic Net	0.734	0.976	0.859	0.956	0.978	0.740
SVM Linear	0.690	0.972	0.839	0.949	0.974	0.704
SVM Radial	**0.732**	**0.976**	**0.855**	**0.956**	**0.980**	**0.730**
XGBoost + Log Reg Elastic Net + SVM Linear	0.664	0.972	0.781	0.949	0.988	0.574
XGBoost Ridge + Log Reg + SVM Radial	0.618	0.969	0.739	0.943	0.990	0.488
XGBoost + Log Reg + SVM Radial	0.618	0.969	0.738	0.943	0.990	0.487

Table B.18: Average results from neural networks when considering *fever* as an auxiliary variable and excluding it from the training data (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.712	0.974	0.849	0.952	0.976	0.723
Deeper NN (more hidden layers)	0.727	0.975	0.860	0.954	0.976	0.744
Smaller NN	0.700	0.973	0.842	0.950	0.976	0.709
Wider NN (more neurons per layer)	0.727	0.975	0.860	0.954	0.976	0.743
NN with batch normalization	0.713	0.973	0.861	0.951	0.972	0.751
NN with LeakyReLU	0.721	0.974	0.855	0.954	0.976	0.734

Table B.19: Average results when considering *night sweats* as an auxiliary variable and excluding it from the training data (inclusion probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.547	0.912	0.775	0.867	0.898	0.651
Log Reg Lasso	0.716	0.966	0.846	0.937	0.967	0.726
Log Reg Ridge	0.697	0.964	0.827	0.933	0.969	0.685
Log Reg Elastic Net	0.714	0.966	0.845	0.934	0.963	0.727
Decision Tree	0.675	0.961	0.828	0.931	0.965	0.690
Random Forest	**0.770**	**0.972**	**0.880**	**0.951**	**0.974**	**0.786**
XGBoost	**0.789**	**0.974**	**0.891**	**0.955**	**0.976**	**0.807**
XGBoost L1	0.748	0.969	0.869	0.946	0.972	0.765
XGBoost L2	0.785	0.974	0.889	0.954	0.975	0.802
XGBoost Elastic Net	0.776	0.972	0.884	0.952	0.974	0.794
SVM Linear	0.731	0.966	0.866	0.941	0.967	0.765
SVM Radial	**0.773**	**0.972**	**0.883**	**0.951**	**0.974**	**0.793**
XGBoost + Log Reg Elastic Net + SVM Linear	0.717	0.968	0.817	0.943	0.985	0.649
XGBoost Ridge + Log Reg + SVM Radial	0.663	0.963	0.774	0.934	0.988	0.560
XGBoost + Log Reg + SVM Radial	0.665	0.964	0.774	0.935	0.988	0.561

Table B.20: Average results from neural networks when considering *night sweats* as an auxiliary variable and excluding it from the training data (oversample probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.756	0.970	0.876	0.947	0.971	0.780
Deeper NN (more hidden layers)	0.766	0.971	0.881	0.949	0.972	0.790
Smaller NN	0.745	0.968	0.872	0.945	0.969	0.775
Wider NN (more neurons per layer)	0.767	0.971	0.880	0.950	0.973	0.787
NN with batch normalization	0.756	0.969	0.883	0.946	0.967	0.800
NN with LeakyReLU	0.763	0.971	0.880	0.949	0.972	0.787

Table B.21: Average results when considering *lack of appetite* as an auxiliary variable and excluding it from training data (inclusion probability=0.34)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.549	0.915	0.774	0.869	0.901	0.648
Log Reg Lasso	0.707	0.965	0.842	0.938	0.970	0.714
Log Reg Ridge	0.691	0.964	0.827	0.936	0.972	0.681
Log Reg Elastic Net	0.706	0.965	0.842	0.938	0.970	0.715
Decision Tree	0.675	0.961	0.828	0.931	0.966	0.691
Random Forest	**0.756**	**0.970**	**0.873**	**0.948**	**0.972**	**0.774**
XGBoost	**0.782**	**0.973**	**0.888**	**0.953**	**0.975**	**0.802**
XGBoost L1	0.745	0.969	0.866	0.946	0.972	0.760
XGBoost L2	0.779	0.973	0.886	0.953	0.975	0.797
XGBoost Elastic Net	0.771	0.972	0.881	0.951	0.974	0.789
SVM Linear	0.726	0.966	0.865	0.940	0.965	0.765
SVM Radial	**0.765**	**0.971**	**0.879**	**0.949**	**0.973**	**0.786**
XGBoost + Log Reg Elastic Net + SVM Linear	0.707	0.967	0.812	0.941	0.985	0.640
XGBoost Ridge + Log Reg + SVM Radial	0.664	0.963	0.773	0.934	0.987	0.558
XGBoost + Log Reg + SVM Radial	0.665	0.963	0.773	0.934	0.987	0.558

Table B.22: Average results from neural networks when considering *lack of appetite* as an auxiliary variable and excluding it from the training data (oversample probability=0.34, N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.749	0.969	0.874	0.946	0.970	0.777
Deeper NN (more hidden layers)	0.764	0.971	0.881	0.949	0.971	0.791
Smaller NN	0.735	0.967	0.867	0.942	0.967	0.767
Wider NN (more neurons per layer)	0.765	0.971	0.881	0.949	0.972	0.789
NN with batch normalization	0.753	0.969	0.880	0.946	0.967	0.793
NN with LeakyReLU	0.757	0.970	0.877	0.947	0.971	0.782

Table B.23: Average results when considering *weight loss and time passed* as auxiliary variables and excluding them from the training data (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.233	0.926	0.585	0.876	0.935	0.236
Log Reg Lasso	0.222	0.955	0.539	0.795	0.848	0.230
Log Reg Ridge	0.217	0.955	0.540	0.777	0.825	0.254
Log Reg Elastic Net	0.222	0.955	0.540	0.794	0.846	0.234
Decision Tree	0.275	0.954	0.570	0.913	0.983	0.157
Random Forest	**0.294**	**0.957**	**0.584**	**0.919**	**0.987**	**0.182**
XGBoost	**0.368**	**0.955**	**0.653**	**0.917**	**0.971**	**0.336**
XGBoost L1	0.305	0.955	0.606	0.916	0.980	0.233
XGBoost L2	0.362	0.955	0.648	0.917	0.972	0.323
XGBoost Elastic Net	0.339	0.955	0.630	0.916	0.975	0.284
SVM Linear	0.194	0.954	0.536	0.914	0.991	0.081
SVM Radial	**0.304**	**0.955**	**0.583**	**0.916**	**0.984**	**0.182**
XGBoost + Log Reg Elastic Net + SVM Linear	0.226	0.956	0.524	0.917	0.997	0.050
XGBoost Ridge + Log Reg + SVM Radial	0.254	0.956	0.539	0.917	0.995	0.083
XGBoost + Log Reg + SVM Radial	0.254	0.957	0.539	0.917	0.995	0.084

Table B.24: Average results from neural networks when considering *weight loss and time passed* as auxiliary variables and excluding them from the training data (N=1000)

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.308	0.955	0.609	0.915	0.978	0.239
Deeper NN (more hidden layers)	0.352	0.954	0.645	0.914	0.969	0.322
Smaller NN	0.291	0.955	0.596	0.915	0.980	0.211
Wider NN (more neurons per layer)	0.350	0.953	0.648	0.913	0.968	0.328
NN with batch normalization	0.369	0.951	0.672	0.910	0.958	0.386
NN with LeakyReLU	0.340	0.954	0.635	0.915	0.972	0.297

Appendix C

The Boxplots of the Simulation Results

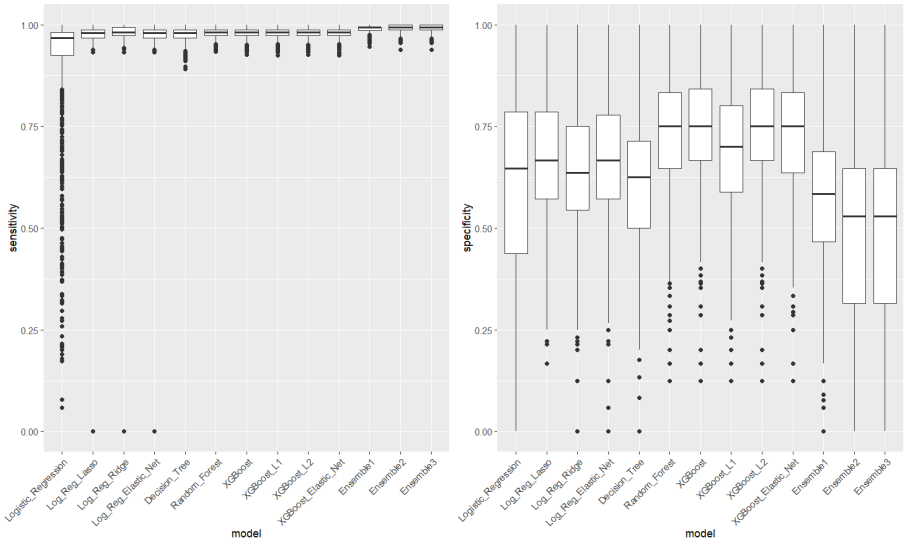


Figure C.1: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable.

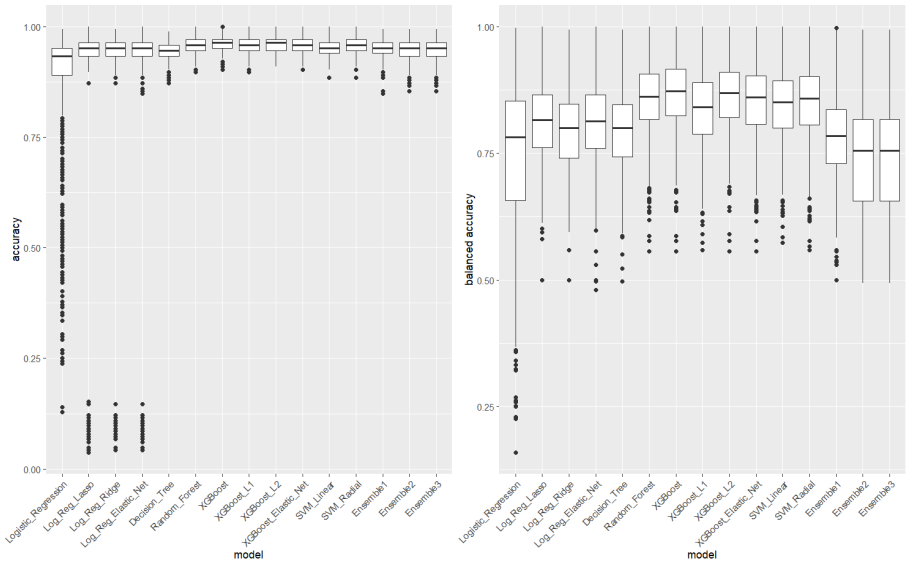


Figure C.2: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable.

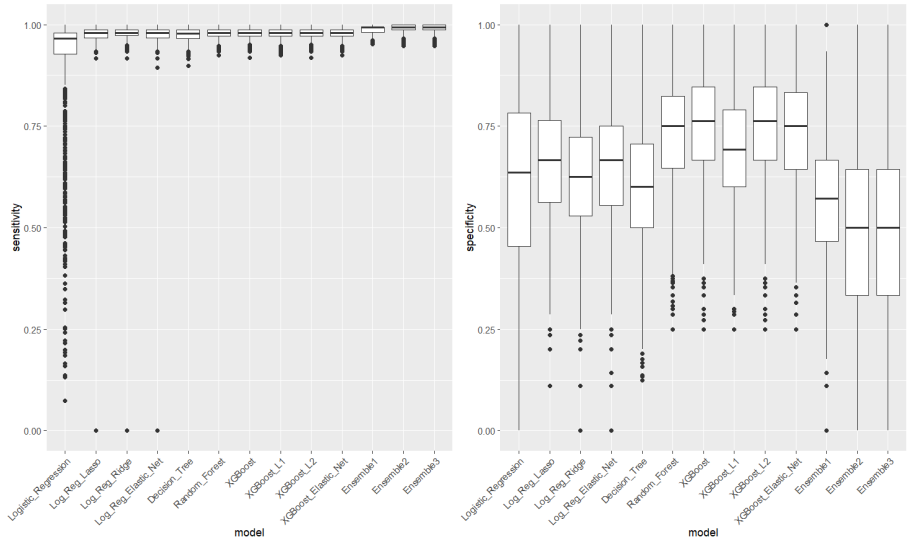


Figure C.3: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable.

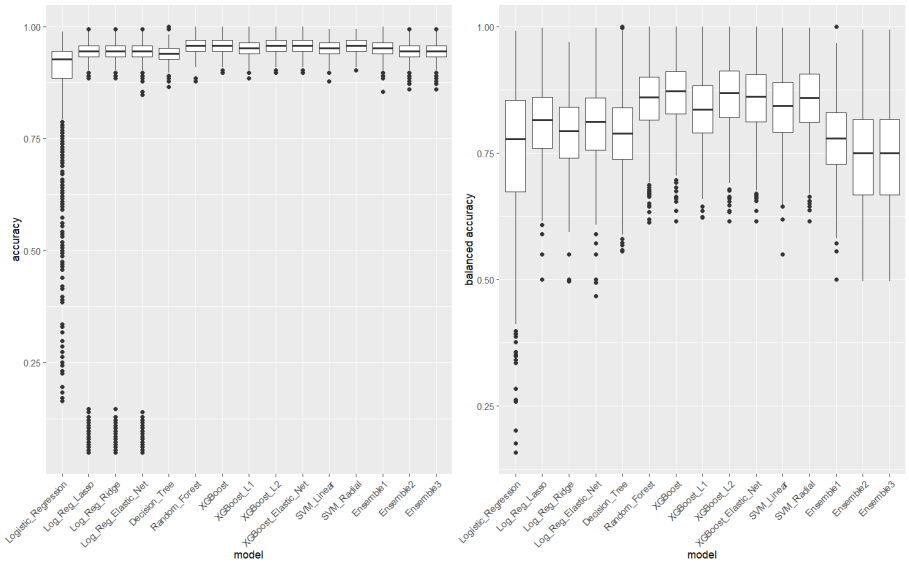


Figure C.4: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable.

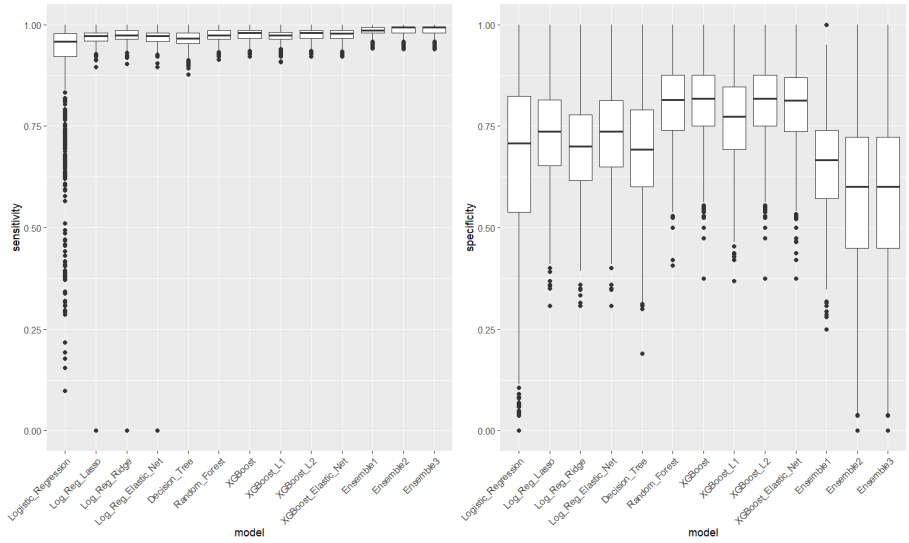


Figure C.5: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable.

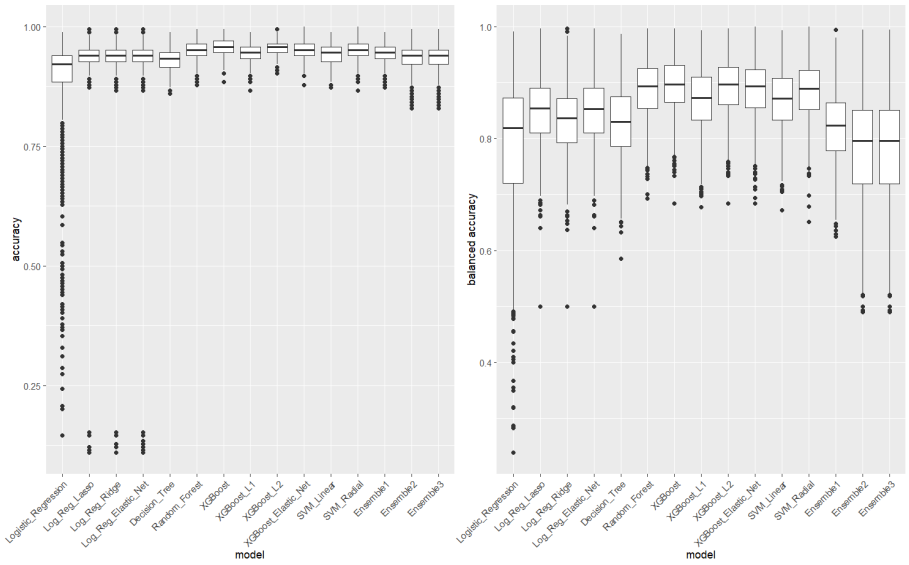


Figure C.6: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable.

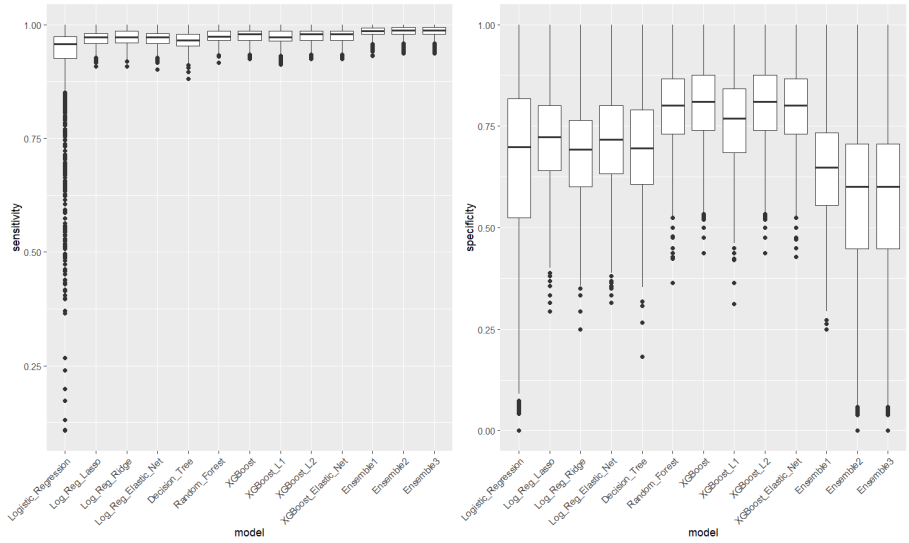


Figure C.7: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable.

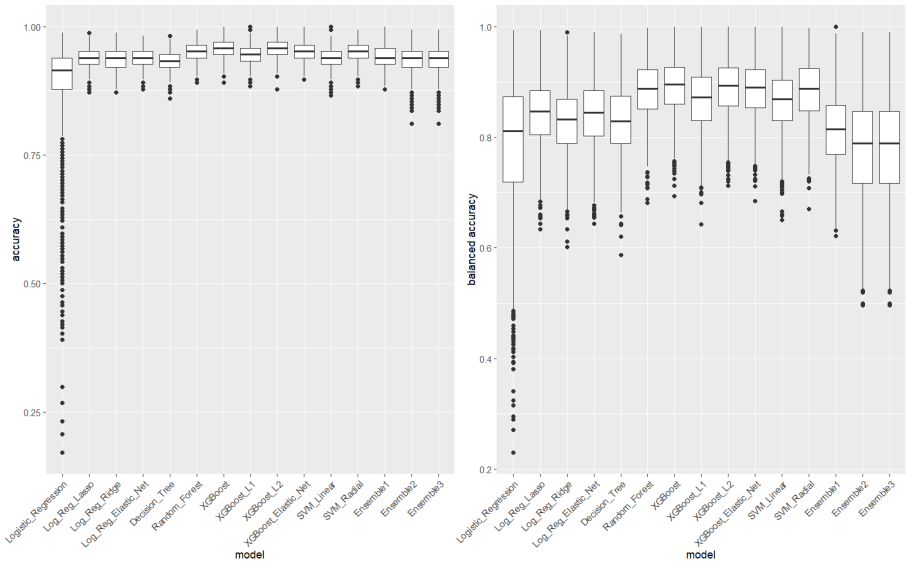


Figure C.8: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable.

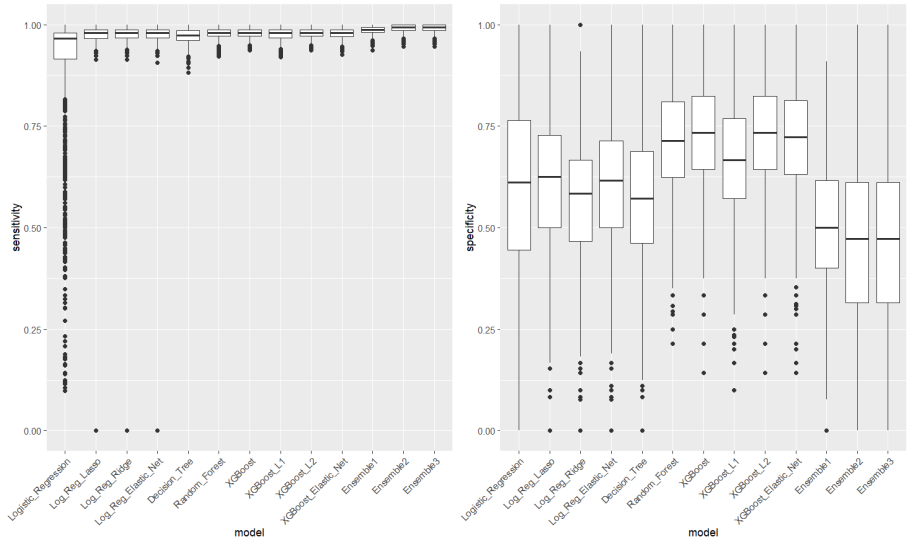


Figure C.9: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

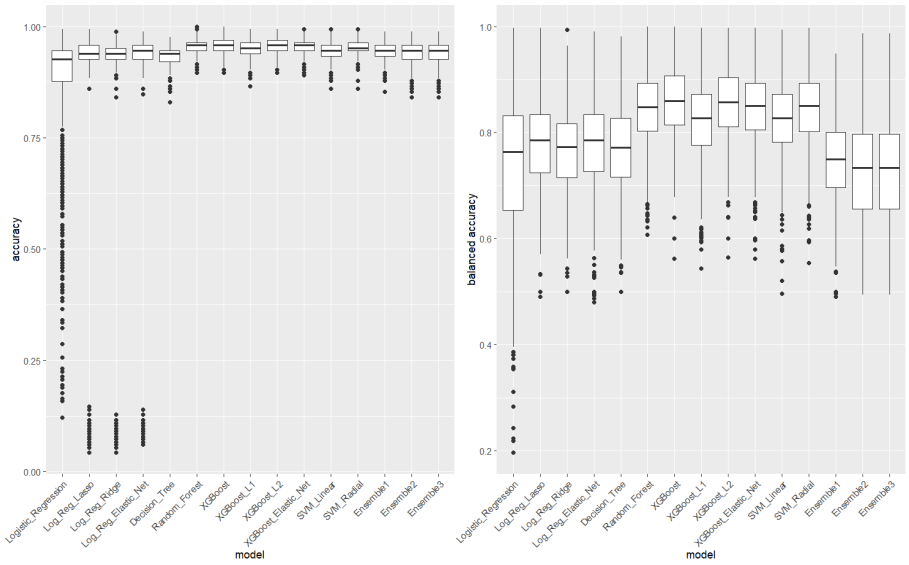


Figure C.10: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

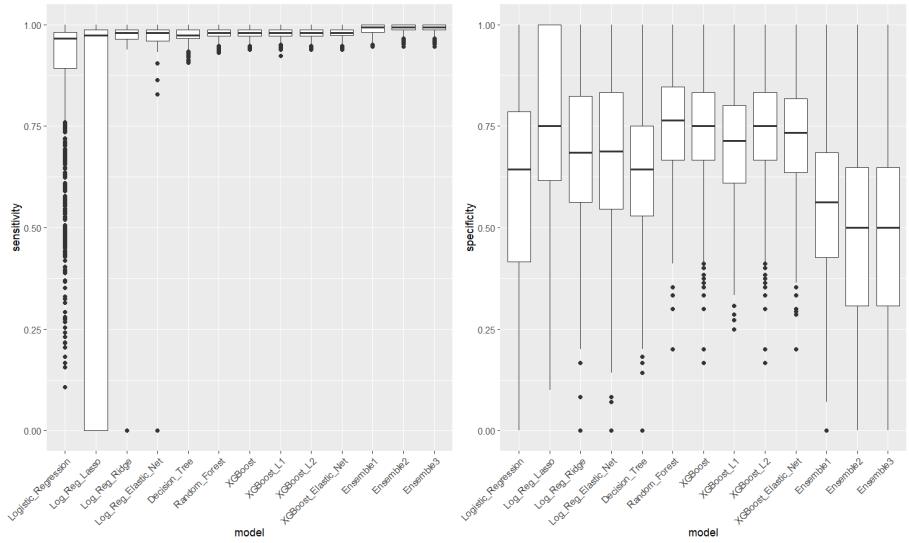


Figure C.11: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables.

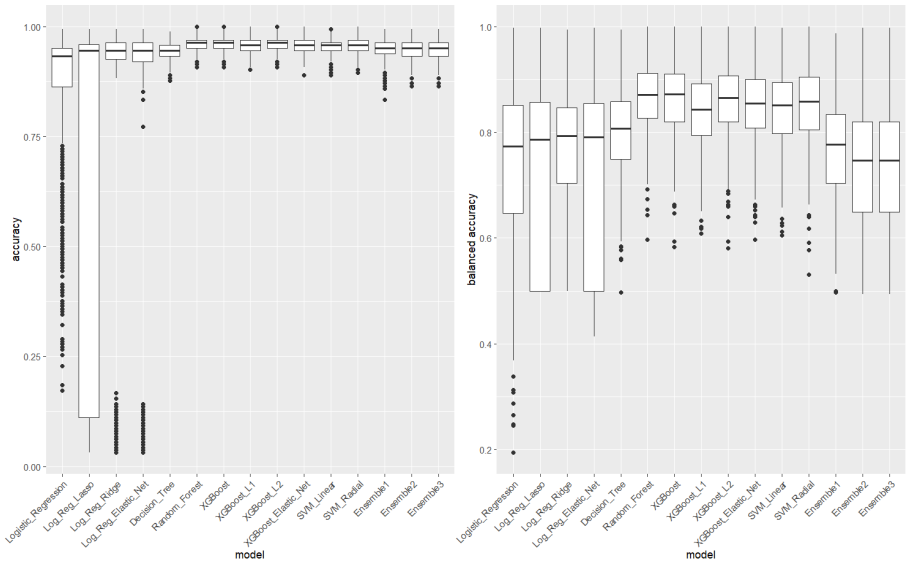


Figure C.12: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables.

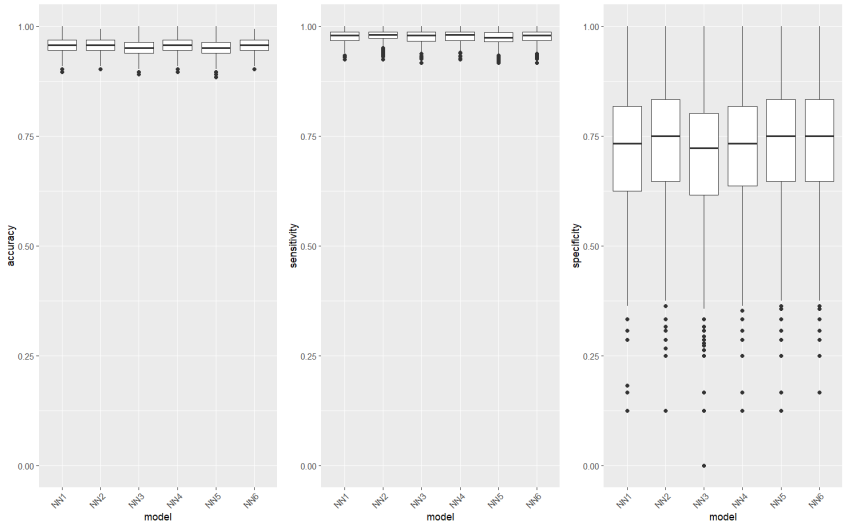


Figure C.13: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable.

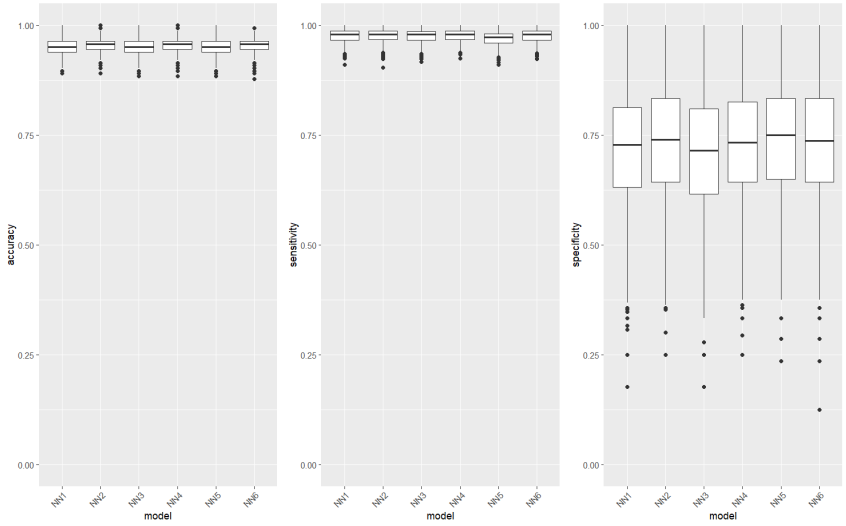


Figure C.14: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable.

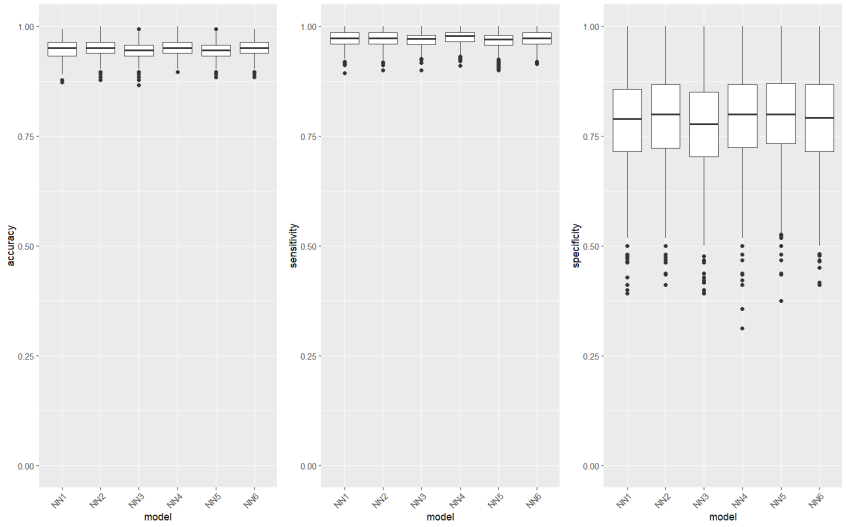


Figure C.15: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable.

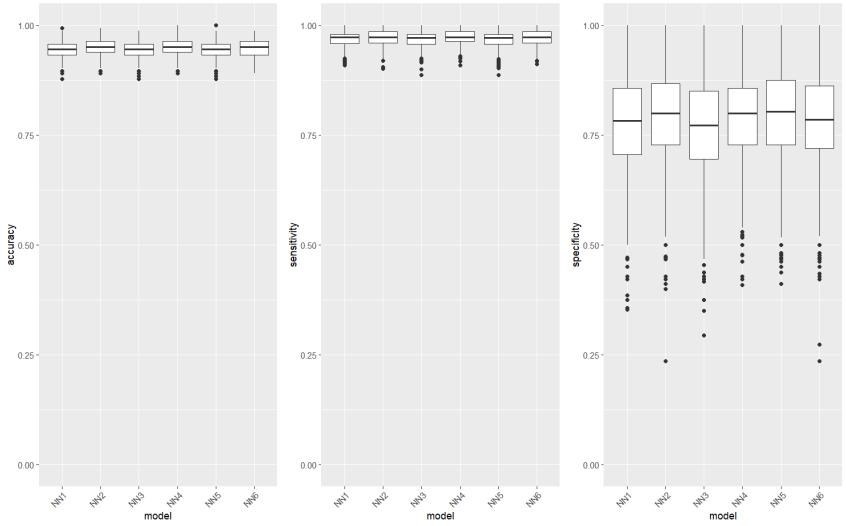


Figure C.16: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable.

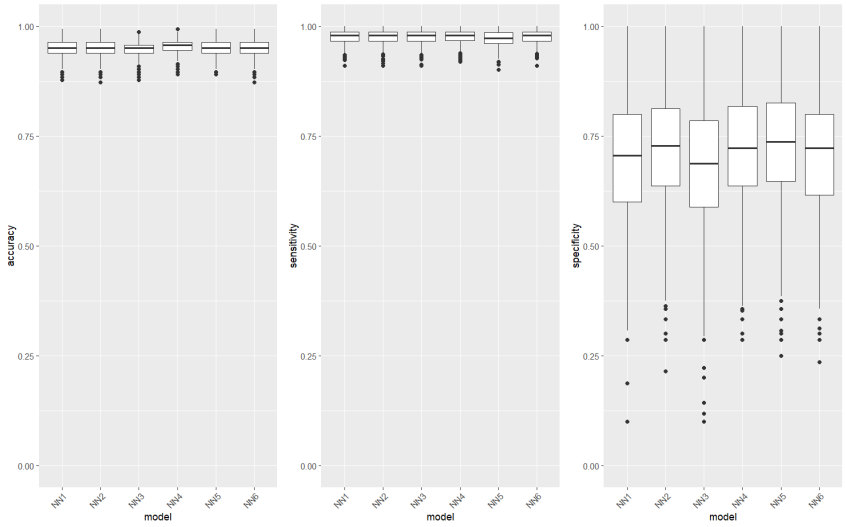


Figure C.17: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable.

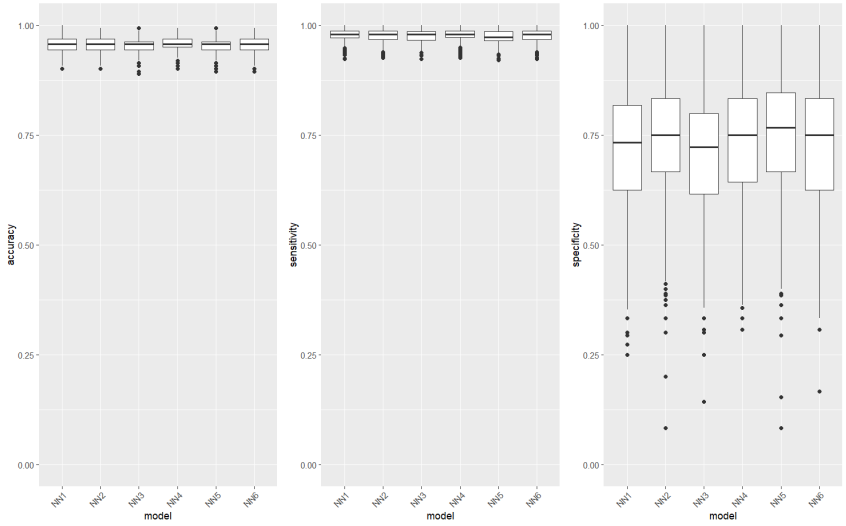


Figure C.18: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variable.

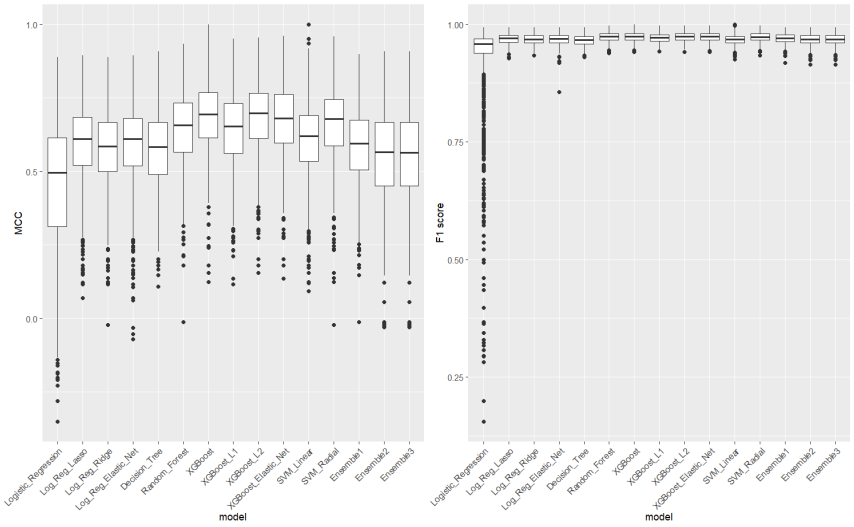


Figure C.19: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data.

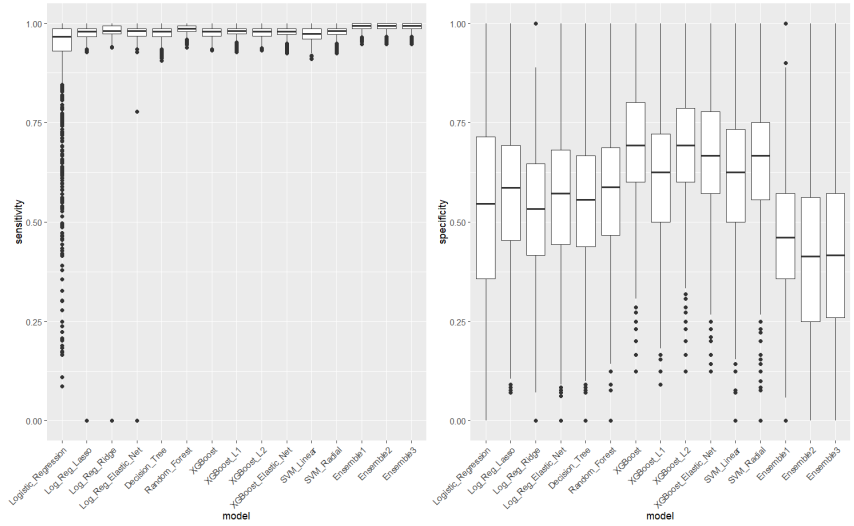


Figure C.20: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data.

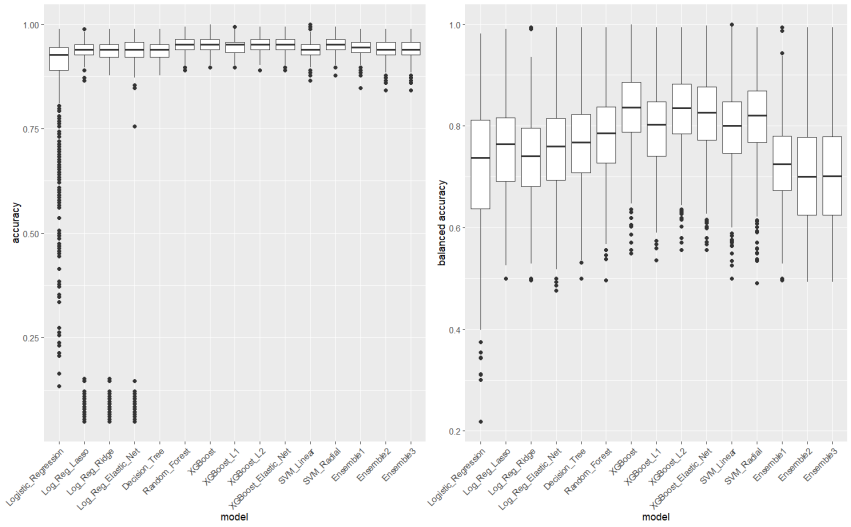


Figure C.21: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data.

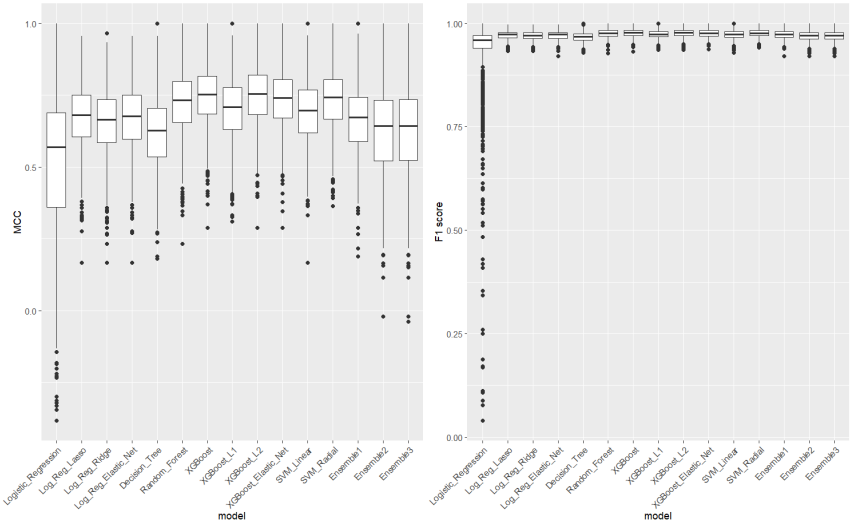


Figure C.22: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data.

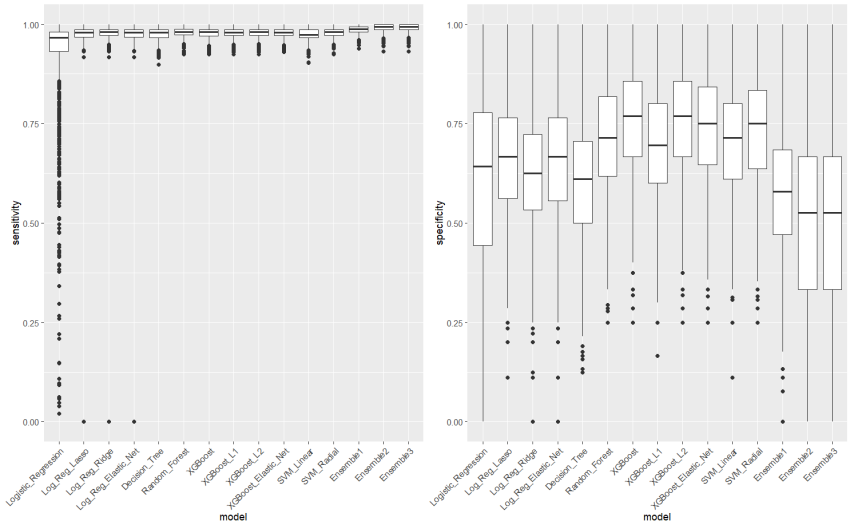


Figure C.23: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data.

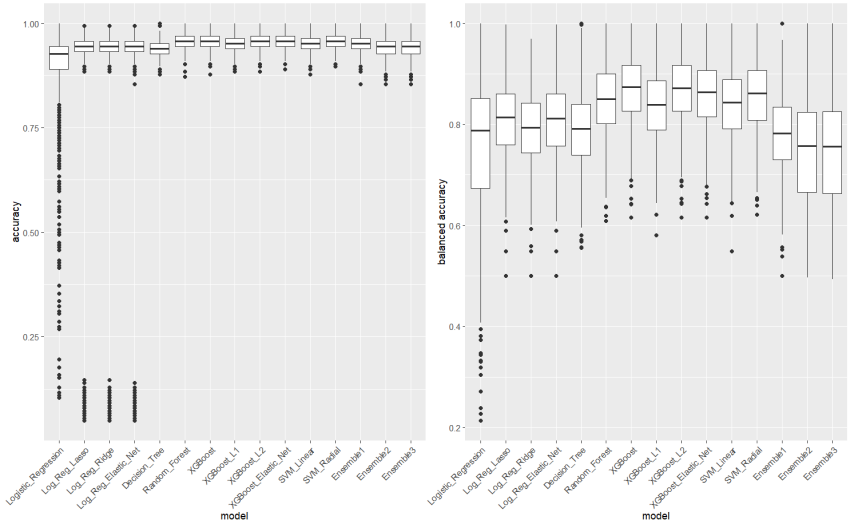


Figure C.24: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data.

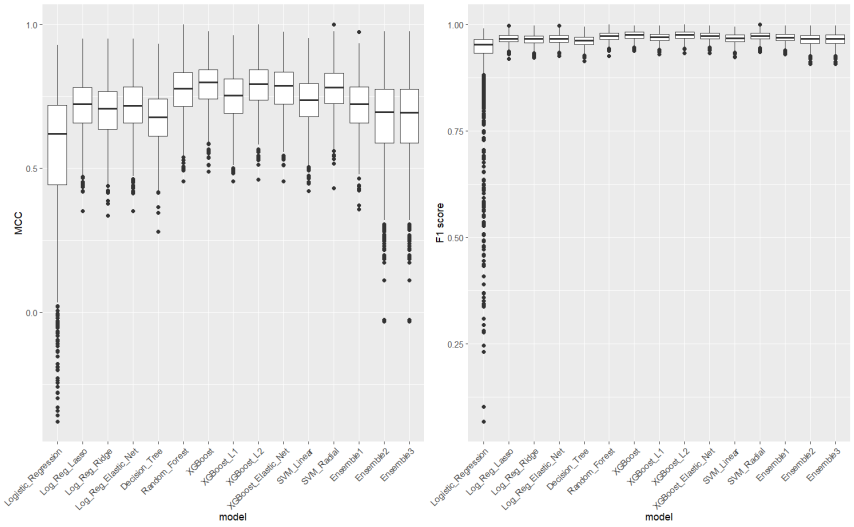


Figure C.25: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data.

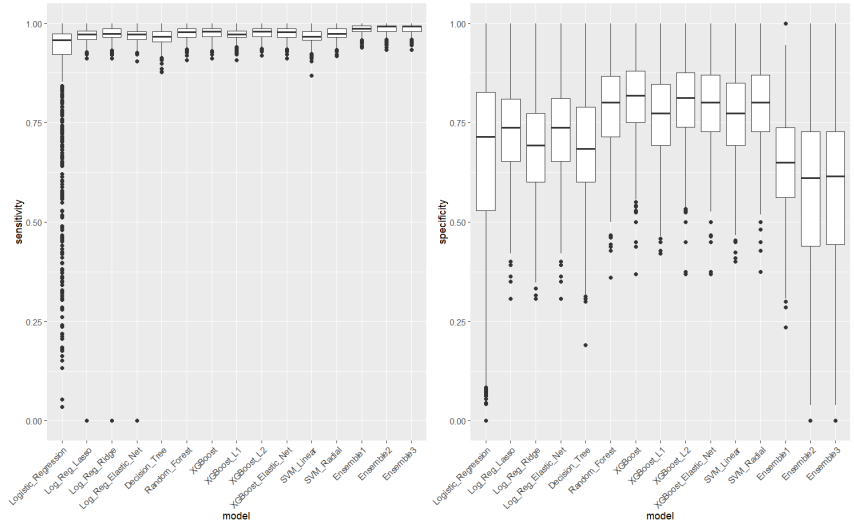


Figure C.26: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data.

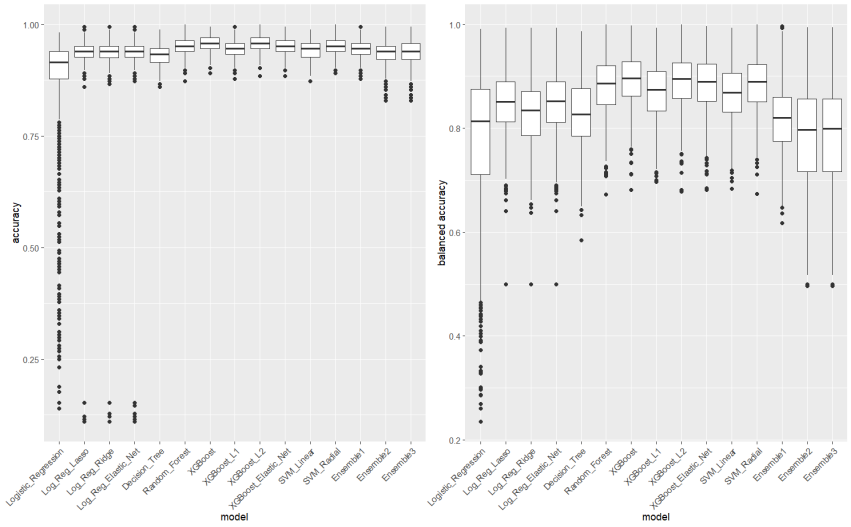


Figure C.27: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data.

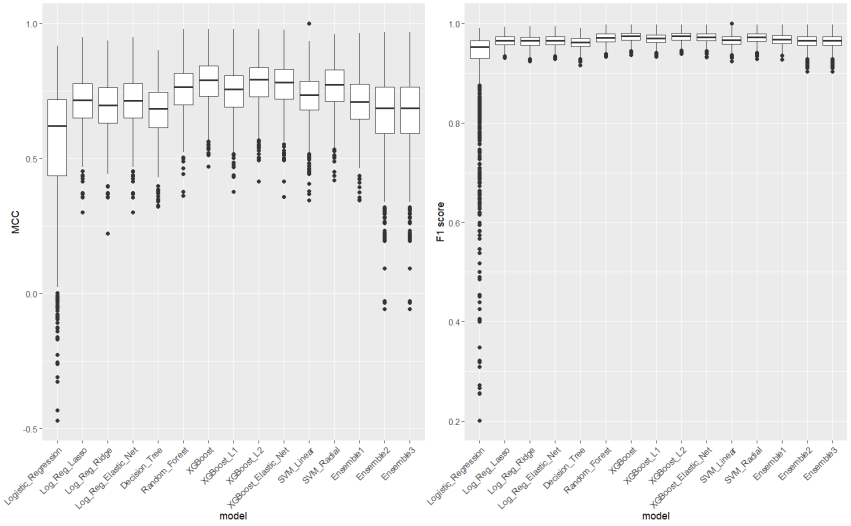


Figure C.28: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data.

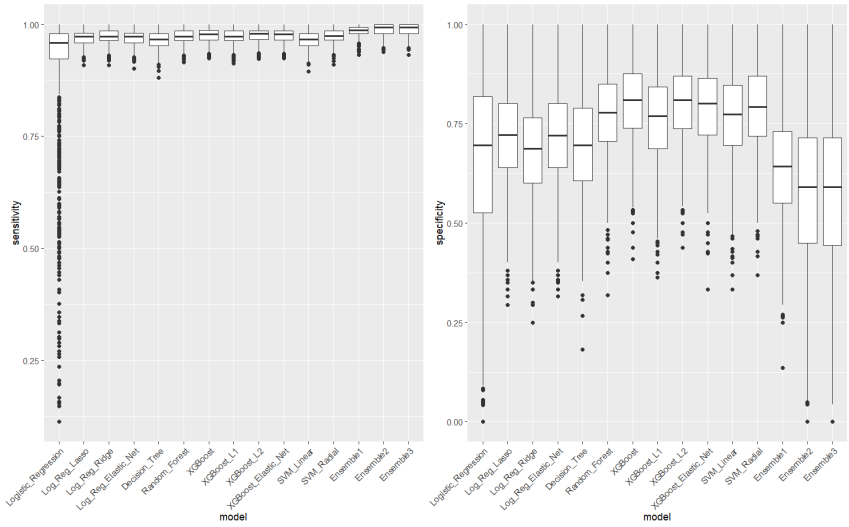


Figure C.29: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data.

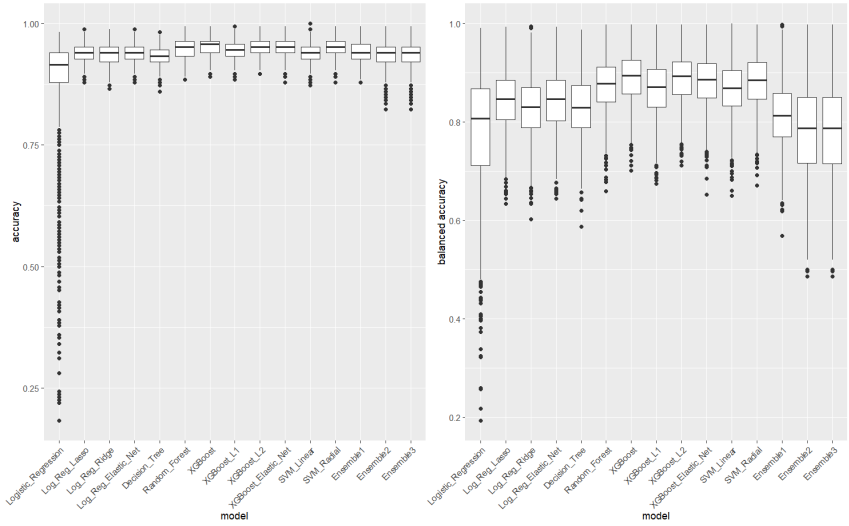


Figure C.30: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data.

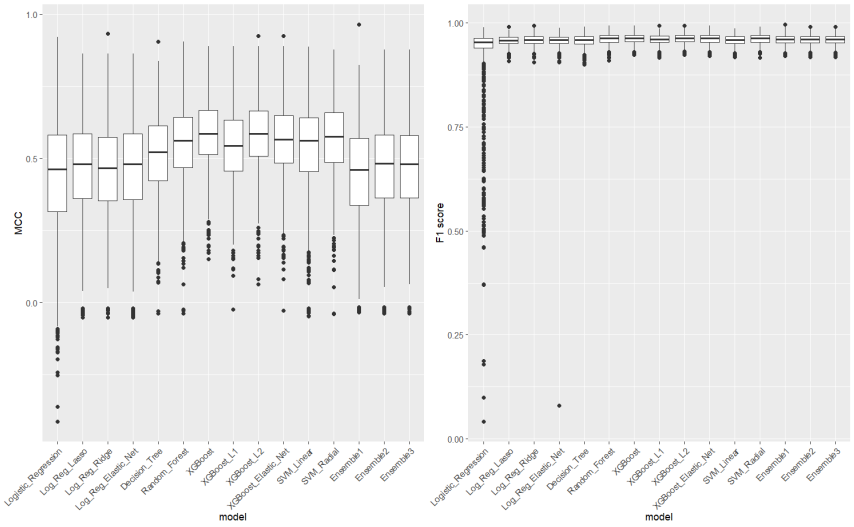


Figure C.31: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data.

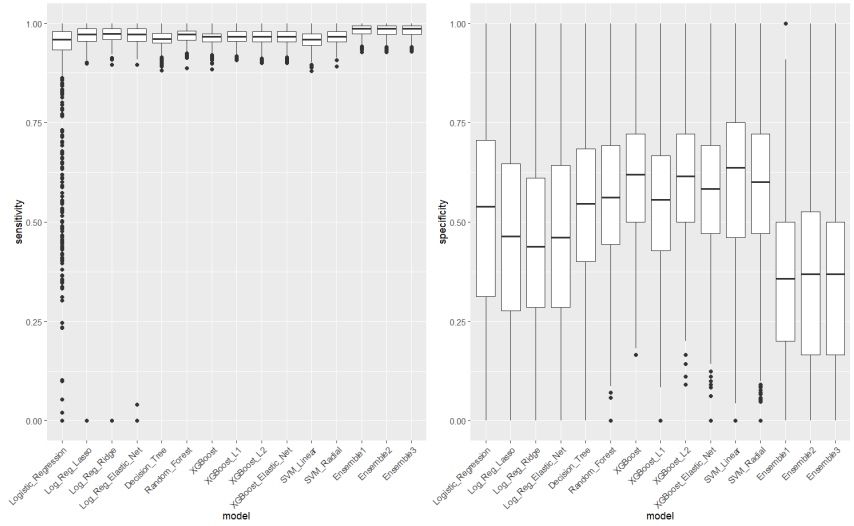


Figure C.32: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data.

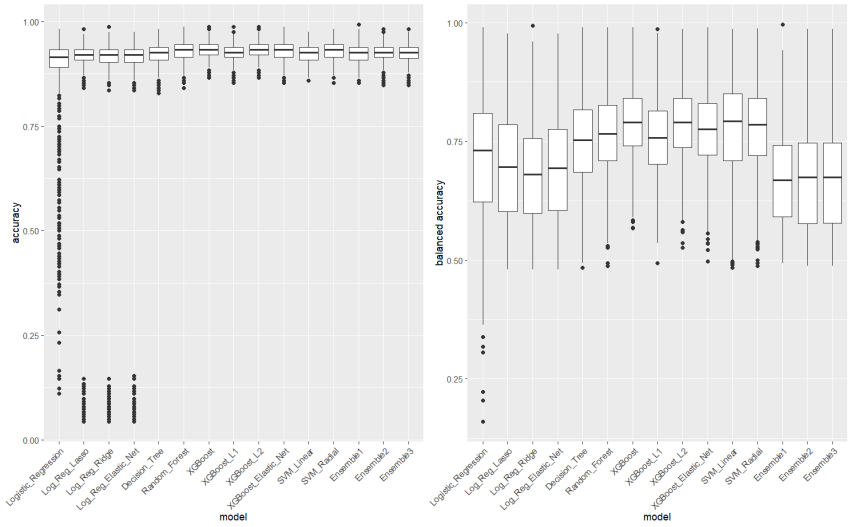


Figure C.33: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data.

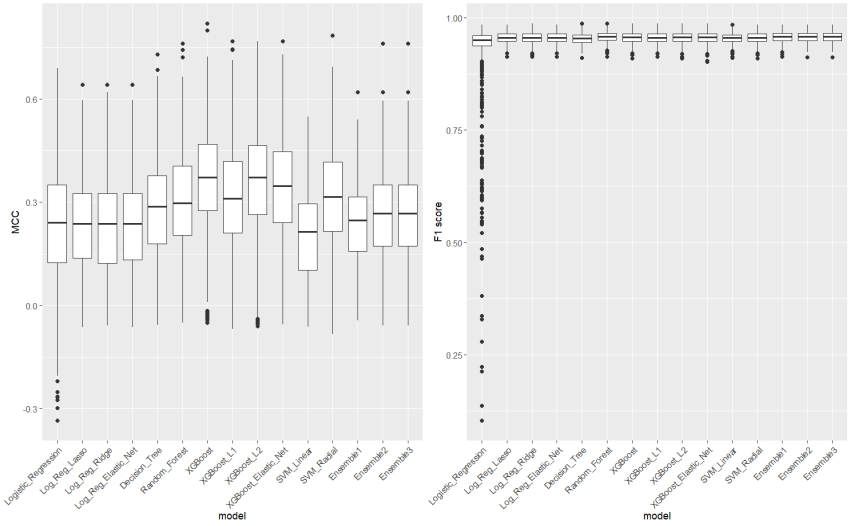


Figure C.34: The boxplots of MCC (left) and F1 score (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data.

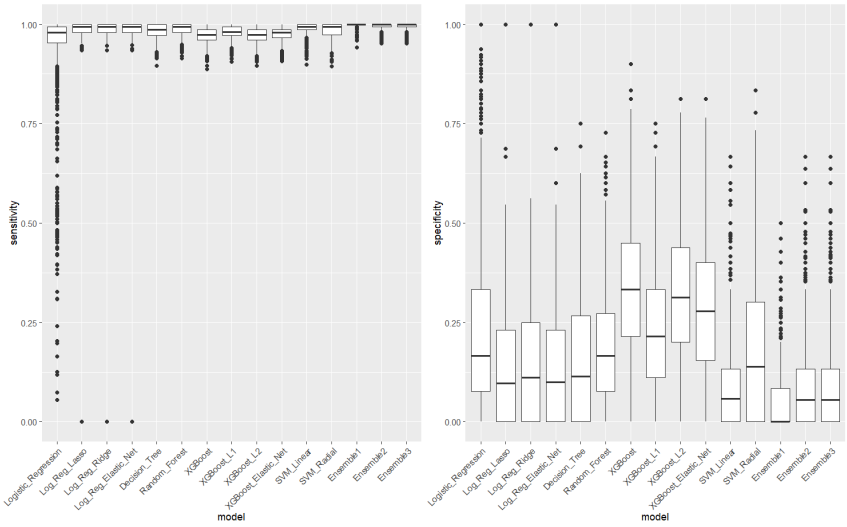


Figure C.35: The boxplots of Sensitivity (left) and Specificity (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data.

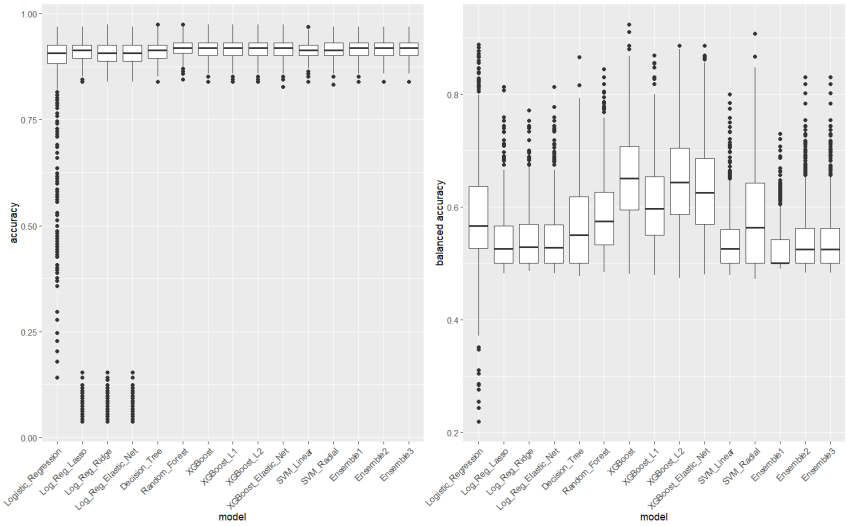


Figure C.36: The boxplots of Accuracy (left) and Balanced accuracy (right) from N=1000 simulated datasets when treating both *weight loss and time passed* as the auxiliary variables and excluding them from the training data.

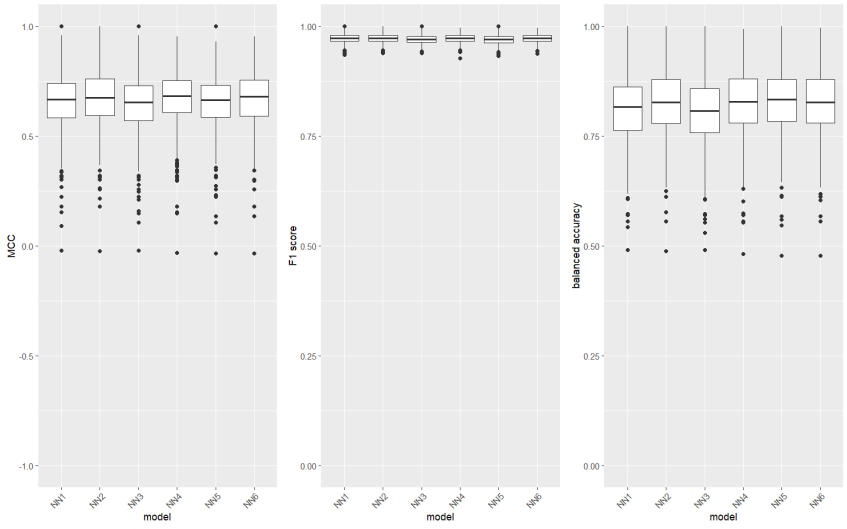


Figure C.37: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data.

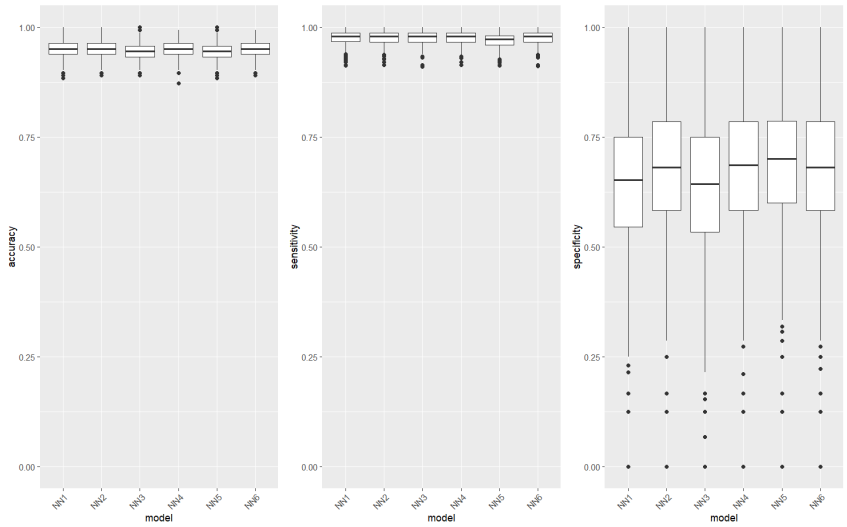


Figure C.38: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* as the auxiliary variable and excluding it from the training data.

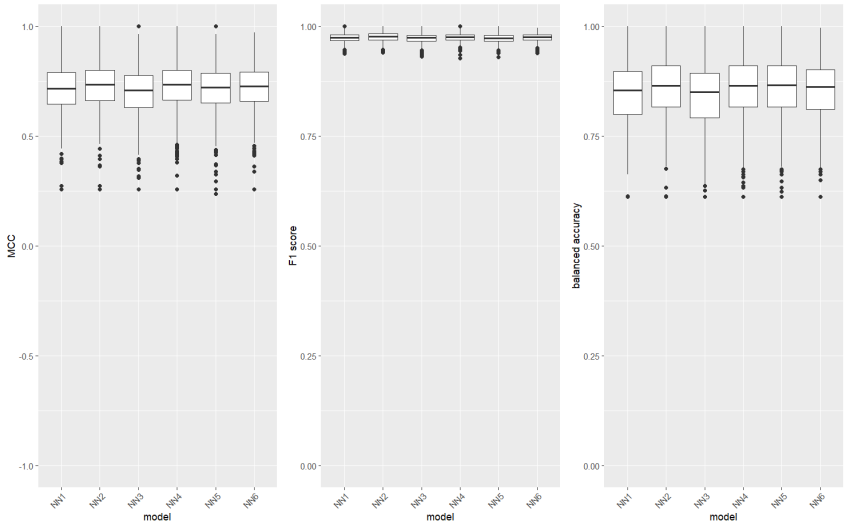


Figure C.39: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data.

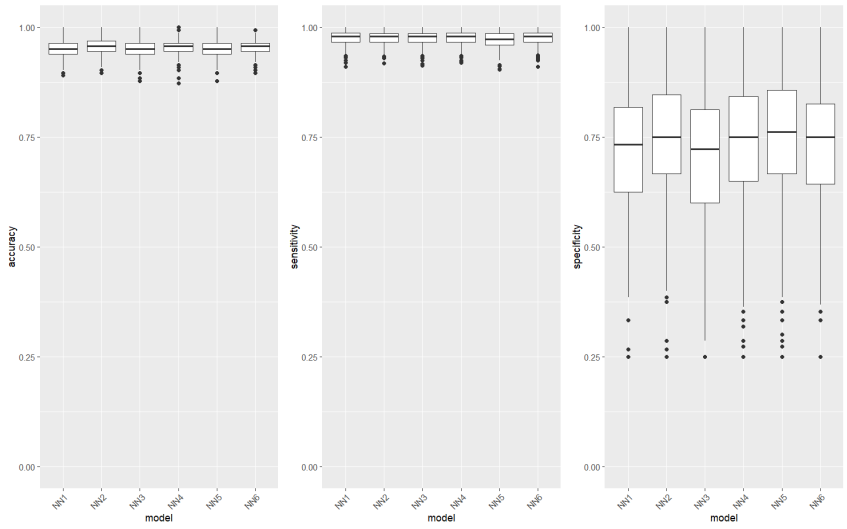


Figure C.40: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *fever* as the auxiliary variable and excluding it from the training data.

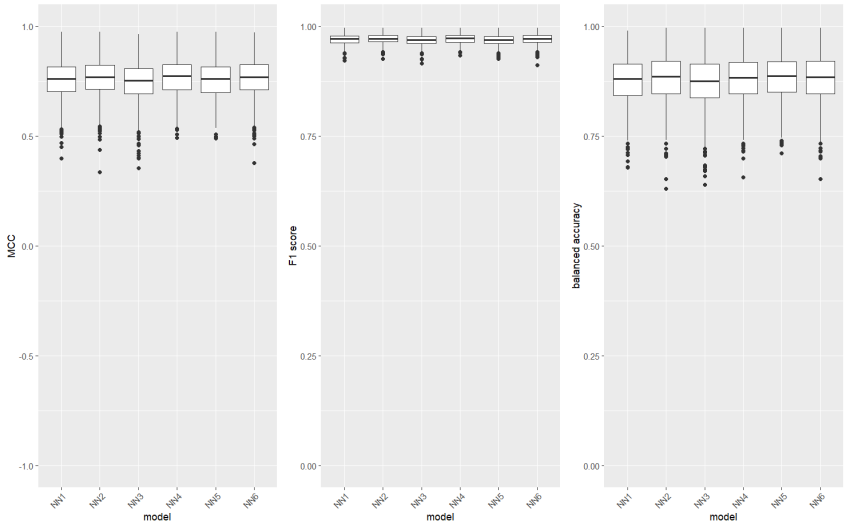


Figure C.41: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data.

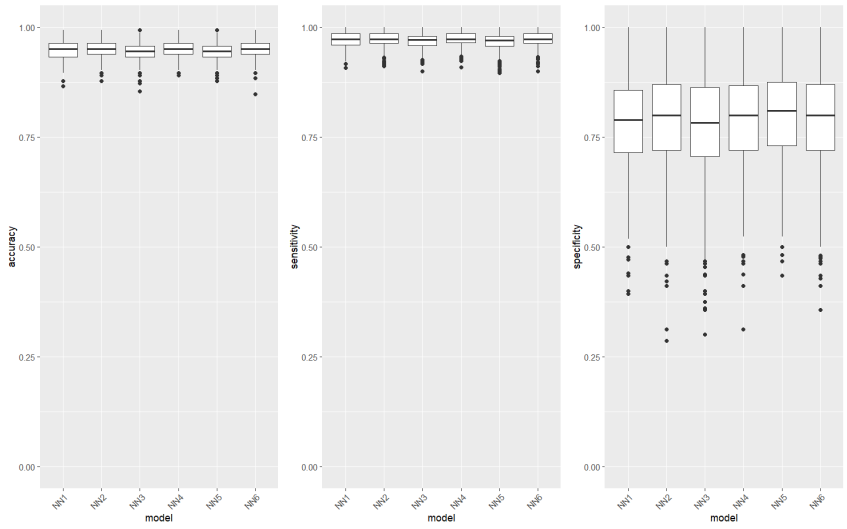


Figure C.42: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *night sweats* as the auxiliary variable and excluding it from the training data.

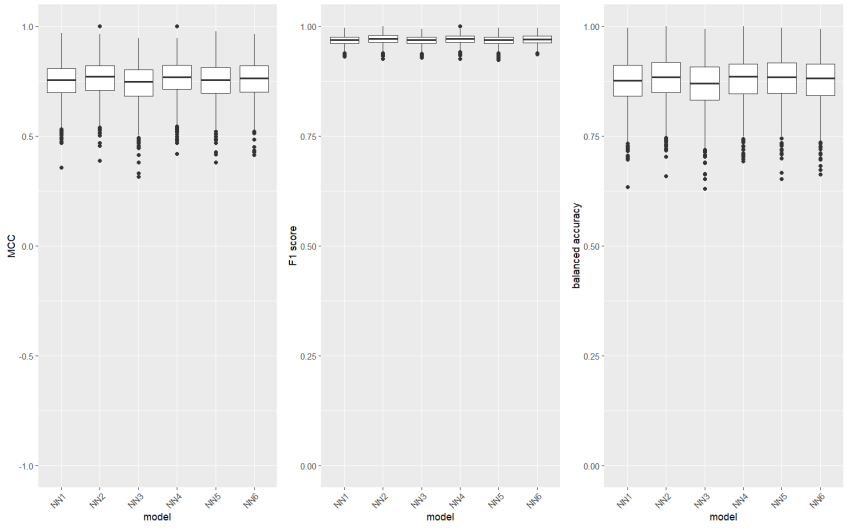


Figure C.43: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data.

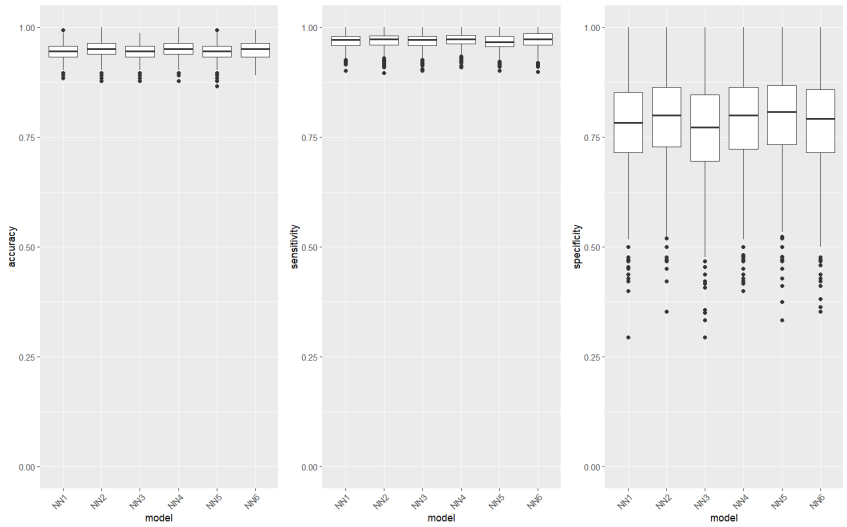


Figure C.44: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *lack of appetite* as the auxiliary variable and excluding it from the training data.

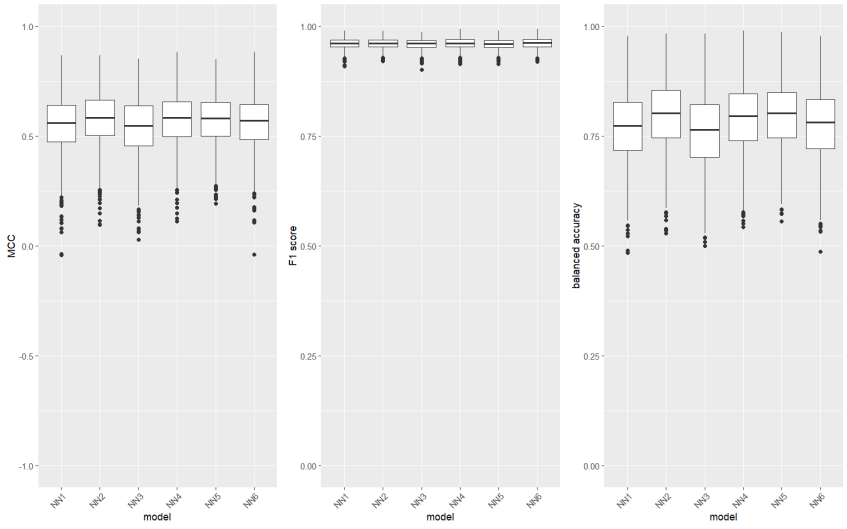


Figure C.45: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data.

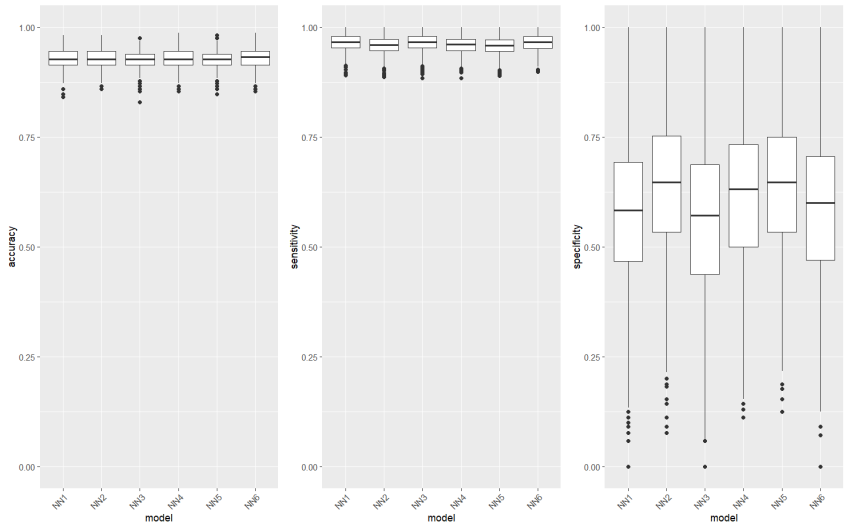


Figure C.46: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *time passed* as the auxiliary variable and excluding it from the training data.

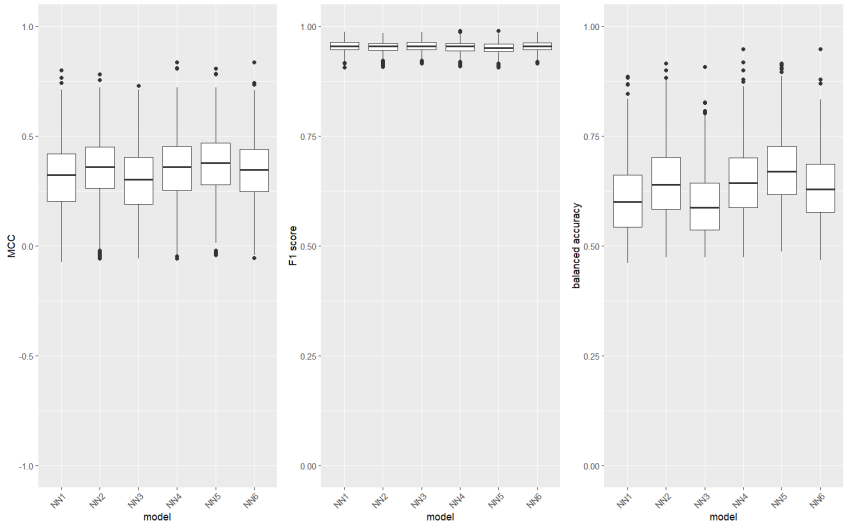


Figure C.47: The boxplots of MCC (left), F1 score (center) and balanced accuracy (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variables and excluding them from the training data.

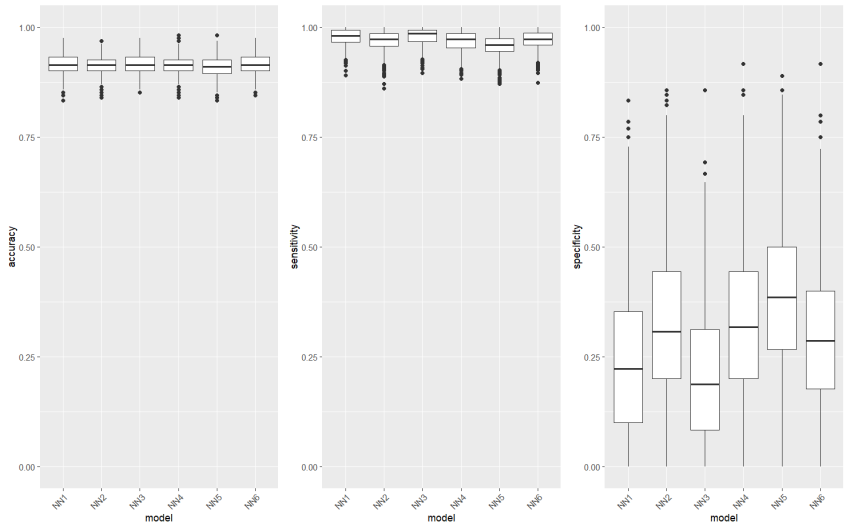


Figure C.48: The boxplots of accuracy (left), sensitivity (center) and specificity (right) from applying neural networks to N=1000 simulated datasets when treating *weight loss* and *time passed* as the auxiliary variables and excluding them from the training data.

Appendix D

Additional Results from the Real-Life Data

Table D.1: Results from the ML models on the real-life data with *night sweats* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.752	0.973	0.895	0.951	0.966	0.824
Log Reg Lasso	0.765	0.976	0.872	0.957	0.980	0.765
Log Reg Ridge	0.725	0.973	0.843	0.951	0.980	0.706
Log Reg Elastic Net	0.765	0.976	0.872	0.957	0.980	0.765
Decision Tree	0.655	0.966	0.810	0.939	0.973	0.647
Random Forest	0.697	0.969	0.839	0.945	0.973	0.706
XGBoost	0.737	0.973	0.869	0.951	0.973	0.765
XGBoost L1	0.612	0.963	0.781	0.933	0.973	0.588
XGBoost L2	0.697	0.969	0.839	0.945	0.973	0.706
XGBoost Elastic Net	0.737	0.973	0.869	0.951	0.973	0.765
SVM Linear	0.752	0.973	0.895	0.951	0.966	0.824
SVM Radial	0.725	0.973	0.843	0.951	0.980	0.706
XGBoost + Log Reg Elastic Net + SVM Linear	0.752	0.973	0.895	0.951	0.966	0.824
XGBoost Ridge + Log Reg + SVM Radial	0.725	0.973	0.843	0.951	0.980	0.706
XGBoost + Log Reg + SVM Radial	0.725	0.973	0.843	0.951	0.980	0.706

Table D.2: Results from the neural networks on the real-life data with *night sweats* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.757	0.973	0.917	0.951	0.959	0.875
Deeper NN (more hidden layers)	0.805	0.980	0.924	0.963	0.973	0.875
Smaller NN	0.797	0.976	0.948	0.957	0.959	0.938
Wider NN (more neurons per layer)	0.716	0.969	0.886	0.945	0.959	0.812
NN with batch normalization	0.735	0.969	0.914	0.945	0.953	0.875
NN with LeakyReLU	0.716	0.969	0.886	0.945	0.959	0.812

Table D.3: Results from the ML models on the real-life data with *lack of appetite* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.483	0.862	0.878	0.780	0.757	1.000
Log Reg Lasso	0.651	0.962	0.851	0.933	0.953	0.750
Log Reg Ridge	0.630	0.963	0.823	0.933	0.959	0.688
Log Reg Elastic Net	0.651	0.962	0.851	0.933	0.953	0.750
Decision Tree	0.673	0.966	0.855	0.939	0.959	0.750
Random Forest	0.680	0.970	0.830	0.945	0.973	0.688
XGBoost	0.765	0.976	0.893	0.957	0.973	0.812
XGBoost L1	0.697	0.969	0.858	0.945	0.966	0.750
XGBoost L2	0.805	0.980	0.924	0.963	0.973	0.875
XGBoost Elastic Net	0.780	0.976	0.921	0.957	0.966	0.875
SVM Linear	0.562	0.956	0.789	0.921	0.953	0.625
SVM Radial	0.757	0.973	0.917	0.951	0.959	0.875
XGBoost + Log Reg Elastic Net + SVM Linear	0.651	0.962	0.851	0.933	0.953	0.750
XGBoost Ridge + Log Reg + SVM Radial	0.797	0.976	0.948	0.957	0.959	0.938
XGBoost + Log Reg + SVM Radial	0.797	0.976	0.948	0.957	0.959	0.938

Table D.4: Results from the neural networks on the real-life data with *lack of appetite* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.604	0.948	0.866	0.909	0.919	0.812
Deeper NN (more hidden layers)	0.681	0.947	0.949	0.909	0.899	1.000
Smaller NN	0.654	0.959	0.876	0.927	0.939	0.812
Wider NN (more neurons per layer)	0.757	0.965	0.966	0.939	0.932	1.000
NN with batch normalization	0.685	0.954	0.928	0.921	0.919	0.938
NN with LeakyReLU	0.678	0.958	0.904	0.927	0.932	0.875

Table D.5: Results from the ML models on the real-life data with *time passed* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.612	0.963	0.834	0.933	0.953	0.714
Log Reg Lasso	0.637	0.970	0.808	0.945	0.973	0.643
Log Reg Ridge	0.585	0.967	0.772	0.939	0.973	0.571
Log Reg Elastic Net	0.637	0.970	0.808	0.945	0.973	0.643
Decision Tree	0.435	0.942	0.749	0.896	0.927	0.571
Random Forest	0.688	0.973	0.844	0.951	0.973	0.714
XGBoost	0.827	0.983	0.951	0.970	0.973	0.929
XGBoost L1	0.610	0.967	0.805	0.939	0.967	0.643
XGBoost L2	0.827	0.983	0.951	0.970	0.973	0.929
XGBoost Elastic Net	0.856	0.987	0.954	0.976	0.980	0.929
SVM Linear	0.660	0.970	0.840	0.945	0.967	0.714
SVM Radial	0.827	0.983	0.951	0.970	0.973	0.929
XGBoost + Log Reg Elastic Net + SVM Linear	0.688	0.973	0.844	0.951	0.973	0.714
XGBoost Ridge + Log Reg + SVM Radial	0.827	0.983	0.951	0.970	0.973	0.929
XGBoost + Log Reg + SVM Radial	0.827	0.983	0.951	0.970	0.973	0.929

Table D.6: Results from the neural networks on the real-life data with *time passed* as an auxiliary variable

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.711	0.980	0.827	0.963	0.987	0.667
Deeper NN (more hidden layers)	0.697	0.981	0.788	0.963	0.993	0.583
Smaller NN	0.767	0.984	0.868	0.970	0.987	0.750
Wider NN (more neurons per layer)	0.613	0.974	0.782	0.951	0.980	0.583
NN with batch normalization	0.668	0.974	0.859	0.951	0.967	0.750
NN with LeakyReLU	0.767	0.984	0.868	0.970	0.987	0.750

Table D.7: Results from the ML models on the real-life data with *weight loss and time passed* as the auxiliary variables

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Logistic Regression	0.876	0.994	0.889	0.988	1.000	0.778
Log Reg Lasso	0.876	0.994	0.889	0.988	1.000	0.778
Log Reg Ridge	0.809	0.990	0.833	0.981	1.000	0.667
Log Reg Elastic Net	0.876	0.994	0.889	0.988	1.000	0.778
Decision Tree	0.611	0.981	0.771	0.963	0.987	0.556
Random Forest	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost L1	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost L2	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost Elastic Net	0.876	0.994	0.889	0.988	1.000	0.778
SVM Linear	0.882	0.993	0.941	0.988	0.993	0.889
SVM Radial	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost + Log Reg Elastic Net + SVM Linear	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost Ridge + Log Reg + SVM Radial	0.876	0.994	0.889	0.988	1.000	0.778
XGBoost + Log Reg + SVM Radial	0.876	0.994	0.889	0.988	1.000	0.778

Table D.8: Results from the neural networks on the real-life data with *weight loss and time passed* as the auxiliary variables

	MCC	F1 score	Balanced Accuracy	Accuracy	Sensitivity	Specificity
Base NN	0.740	0.976	0.906	0.957	0.966	0.846
Deeper NN (more hidden layers)	0.664	0.969	0.864	0.944	0.960	0.769
Smaller NN	0.640	0.966	0.861	0.938	0.953	0.769
Wider NN (more neurons per layer)	0.690	0.973	0.868	0.951	0.966	0.769
NN with batch normalization	0.664	0.969	0.864	0.944	0.960	0.769
NN with LeakyReLU	0.690	0.973	0.868	0.951	0.966	0.769

Table D.9: Most important features by auxiliary variable scenario

Auxiliary variable(s)	**Most important features**
None (no bias correction)	marital_status_Widowed, marital_status_Married, marital_status_Single
Weight loss	gender_1, gender_2, marital_status_Divorced
Fever	marital_status_NA, weight_loss_0, time_passed_2
Night sweats	time_passed_1, time_passed_NA, contact_tb_NA
Lack of appetite	marital_status_NA, weight_loss_0, time_passed_1
Time passed	marital_status_NA, weight_loss_0, night_sweats_0
Weight loss, time passed	gender_1, gender_2, marital_status_Divorced