ABSTRACT

Title of Dissertation: COMPARING THE EFFECTIVENESS OF STANDARD VS. MULTILEVEL MACHINE LEARNING ALGORITHMS ON HIERARCHICAL DATA

Brennan Register

Dissertation Directed by: Associate Professor Tracy Sweet
Department of Human Development and
Quantitative Methodology

This dissertation explored the performance of standard and multilevel machine learning classification algorithms on hierarchical datasets, which are prevalent in educational research due to multistage sampling techniques (e.g., students nested within schools). Hierarchical data pose unique analytical challenges (e.g., nonindependence of data), often requiring specialized approaches. While multilevel modeling is well-established in inferential contexts, its potential in predictive scenarios had been largely underexplored.

Through a Monte Carlo simulation and empirical analyses using data from the Maryland Longitudinal Data System (MLDS), this study evaluated the suitability of standard and multilevel algorithms. Results revealed that multilevel models offered slight advantages in high residual level-2 variance settings, effectively capturing cluster-level dependencies and providing stable predictions. Standard models performed well in low residual level-2 variance contexts, while standard models incorporating cluster IDs as fixed effects performed comparably to mul-

tilevel models under many conditions, including high residual level-2 variance scenarios. However, this approach was only feasible when training and testing clusters overlapped, highlighting limitations in generalizing predictions to unseen clusters. In an empirical analysis addressing class imbalance, Logistic Regression and GLMM exhibited the highest sensitivity for identifying STEM completers when training and testing clusters overlapped while Neural Nets and XGBoost demonstrated better performance in identifying the minority class when training and testing clusters were distinct. These findings highlighted the complexity of predictive modeling for hierarchical data and provided insights for prediction tasks in educational research.

COMPARING THE EFFECTIVENESS OF STANDARD VS. MULTILEVEL MACHINE LEARNING ALGORITHMS ON NESTED DATA

by

Brennan Register

Doctoral Dissertation Proposal
January 30, 2024

Advisory Committee:
Associate Professor Tracy Sweet, Chair/Advisor
Professor Gregory R. Hancock
Professor Hong Jiao
Professor Laura Stapleton
Professor Mohammad Hajiaghayi

Acknowledgments

This research was supported by a grant from the American Educational Research Association which receives funds for its "AERA-NSF Grants Program" from the National Science Foundation under NSF award NSF-DRL #1749275. Opinions reflect those of the author and do not necessarily reflect those AERA or NSF.

The author acknowledges the University of Maryland supercomputing resources (http://hpcc.umd.edu) made available for conducting the research reported in this paper.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 The Independence Assumption in Statistical Models

Many statistical and machine learning models rely on the assumption that observations are independent and identically distributed (IID). This assumption means that the outcome of one observation does not influence or depend on another, and that all observations come from the same population. The IID assumption ensures that each observation contributes unique and unbiased information to the analysis, allowing models to accurately estimate relationships between predictors and the outcome. In linear regression, the independence assumption is particularly important because it ensures valid statistical inference, leading to unbiased parameter estimates and accurate standard errors (Casella and Berger, 2002). When data satisfy this assumption, as shown in Figure 1.1, the observations are scattered randomly and independently, with no discernible patterns or groupings. In contrast, Figure 1.2 illustrates a dataset which do not satisfy this assumption.

1.2 Multistage Sampling and Violations of Independence

Educational research frequently uses multistage sampling techniques which are practical and efficient methods for selecting participants from large and geographically dispersed popula-

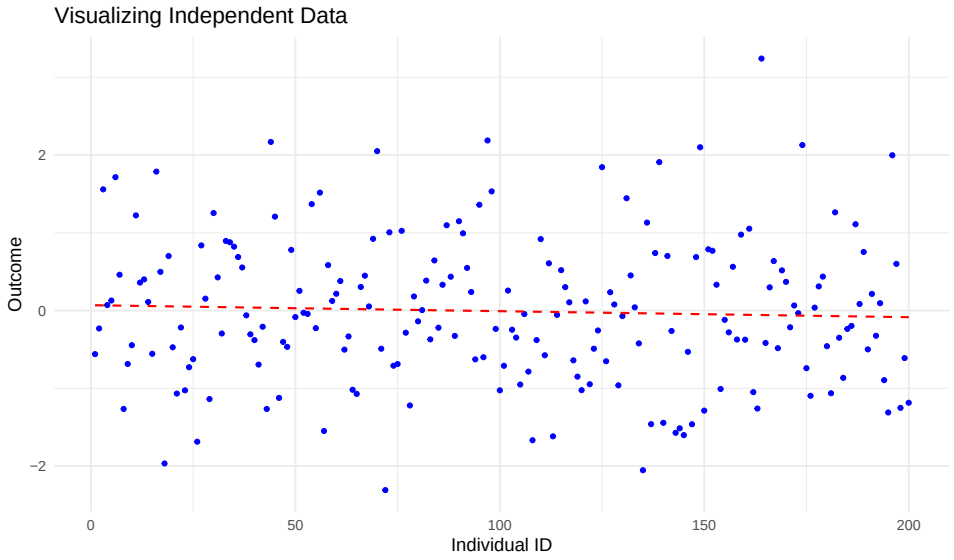

Figure 1.1: Scatter plot illustrating independent and identically distributed (IID) data. Each observation is randomly scattered with no discernible patterns or groupings.

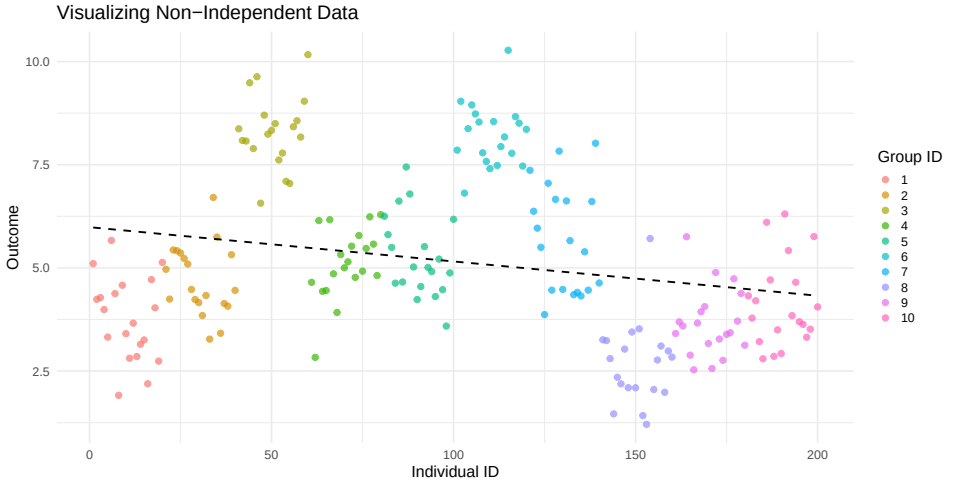

Figure 1.2: Scatter plot illustrating non-independent data where clustering is apparent. Observations within clusters are more similar to each other than to those in different clusters, violating the independence assumption.

tions (Rahman et al., 2022). This method involves selecting samples in stages, beginning with large clusters and progressively narrowing down to smaller units. For instance, in a study assessing the effectiveness of a new teaching method, researchers might begin by selecting school districts as their primary sampling units. They could then choose schools within each district, followed by classrooms or individual students within those schools. While this sampling method may offer logistical and cost advantages, it also introduces specific challenges to statistical modeling such as the violation of the assumption that observations are independent and identically distributed (Muschelli et al., 2014; Rutkowski et al., 2022). Multistage sampling violates this assumption because observations within the same cluster, such as students in a classroom, are often more similar to each other than to those in other clusters. Students in the same classroom, for example, share common influences like the same teacher, curriculum, and peer environment. These shared factors create dependencies among observations which means that observations within a cluster are no longer statistically independent.

1.3 From Multistage Sampling to Hierarchical Data Structures

One of the most significant consequences of multistage sampling is the introduction of clustering in the data, where lower-level units (e.g., students) are nested within higher-level units (e.g., classrooms), which creates a hierarchical structure as illustrated in Figure 1.3.

To quantify the extent of clustering, researchers often use the intraclass correlation coefficient (ICC), which measures the proportion of total variance in the outcome variable attributable to differences between clusters. The ICC ranges from 0 to 1 and provides insight into the degree of similarity among individuals within the same cluster (Chen et al., 2017). When ICC $= 0$, all

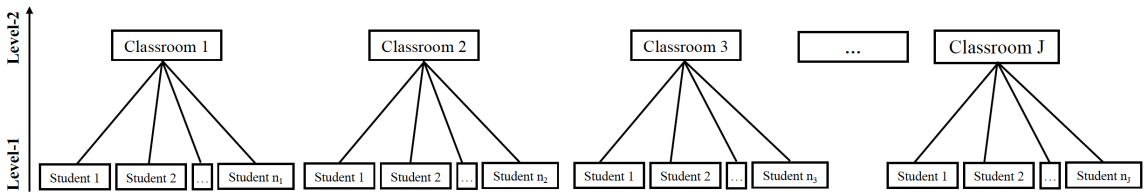

Figure 1.3: Schematic representation of a hierarchical data structure resulting from multistage sampling. Lower-level units (e.g., students) are nested within higher-level units (e.g., classrooms), demonstrating the clustering that characterizes hierarchical data.

variation is within clusters, meaning clusters are homogeneous, with no systematic differences between them. When ICC $= 1$, all the variation is between clusters, meaning the individuals within clusters are identical, but clusters differ completely from one another. When $0 <$ ICC < 1, there is some degree of clustering, where individuals within the same cluster are more similar to one another than to individuals in different clusters. In educational contexts, ICC values are often substantially greater than zero (Snijders and Bosker, 2011). For example, if a study measures student test scores across schools, a high ICC indicates that a lot of the variability in scores arises from differences between schools, such as disparities in resources, teacher quality, or district policies, rather than from differences among students within the same school. Students in well-funded schools, for instance, might consistently outperform those in under-resourced schools, leading to a significant proportion of the total variance being attributed to the school level. When ICC > 0, the independence assumption made by many statistical and machine learning models may not hold. In such cases, alternative modeling approaches, such as multilevel modeling, may be beneficial to account for the hierarchical structure.

1.4 Implications for Predictive Modeling

While the consequences of not using multilevel models in the context of hierarchical data are well-documented in inferential statistics, such as biased parameter estimates and underestimated standard errors (Hox, 1998; Kreft and De Leeuw, 1998; Maas and Hox, 2004), researchers have yet to fully explore how assuming independence affects prediction tasks for hierarchical data. Prediction-focused research does not necessarily seek to explain relationships among variables but rather aims to create models that accurately predict outcomes for new data. When models are trained on hierarchical data without accounting for clustering, they may fail to generalize beyond the clusters present in the training set. Meaning, if dependencies among nested observations are ignored, standard models might perform well in-sample but struggle to generalize to other clusters, such as new classrooms or schools. For instance, a model predicting student achievement that depends heavily on school-level patterns may perform well on schools included in the training set but poorly on new schools with different characteristics. While multilevel machine learning algorithms have been developed to extend standard models to accommodate hierarchical structures, (Fokkema et al., 2018; Fontana et al., 2021; Hajjem et al., 2017; Nguefor et al., 2019; Pellagatti et al., 2021; Speiser et al., 2019, 2020), little research has explored the performance of standard (i.e., single level) classification algorithms compared to their multilevel counterparts on hierarchical data. As machine learning algorithms increasingly inform high-stakes decisions in research and policymaking (Arroway et al., 2015; Engler, 2021), it is important to understand how these methods perform in hierarchical contexts. Therefore, there is a pressing need for additional research to evaluate the feasibility and utility of both standard and multilevel algorithms in hierarchical data settings.

1.5 Scope and Insights of the Study

The dissertation examined the performance of different machine learning algorithms, including both standard and multilevel variants, when applied to multistage sample datasets. Specifically, this research demonstrated how hierarchical data structures influence the predictive accuracy of these algorithms. While the inferential consequences of ignoring the nested structure of data are well-documented, this dissertation focused on prediction accuracy, providing practical insights for researchers working with hierarchical data. Hierarchical data are common in education (e.g., students within schools), healthcare (e.g., patients within hospitals), social sciences (e.g., individuals within families), and longitudinal studies (e.g., repeated observations within individuals). These types of data present unique challenges which can require specialized analytical approaches. Despite the well-established role of multilevel modeling in inferential statistics, the potential of multilevel machine learning algorithms in classification settings remains largely unexplored.

This dissertation used a Monte Carlo simulation to compare the predictive performance of various standard and multilevel algorithms under several conditions including: the complexity of the data-generating model, the amount of residual level-2 variance, the number of clusters, number of individuals per cluster, and the mechanism used to split the data into training and testing sets. A within condition comparison was also conducted to evaluate the impact of including cluster IDs as fixed effects in standard algorithms when the training and testing data are split within cluster.

In addition to the simulation study, empirical analyses were conducted using real-world data from the Maryland Longitudinal Data System (MLDS). These analyses focused on predict-

ing STEM persistence, an important topic in education policy. They also extended the scope of the study by assessing predictive performance in the context of imbalanced data, where one category of an outcome is much more prevalent than the other. Accurate predictions in this context could help inform strategies to support students and improve retention in STEM fields.

This dissertation contributes to the understanding of prediction in hierarchical data settings and offers practical guidance for model selection for educational researchers.

Chapter 2: Theoretical Framework and Literature Review

This chapter lays the foundation for the study by addressing both the theoretical framework and the existing literature. The first section introduces the machine learning algorithms considered in this research, including standard and multilevel approaches, as well as other methods relevant to the reviewed studies. Evaluation metrics commonly used to assess model performance are also presented.

The second section reviews the current applications of machine learning in education research, particularly for predictive modeling. It also examines existing comparisons between standard and multilevel classification algorithms, highlighting gaps in the literature that informed this dissertation's research questions and methodology.

2.1 Theoretical Framework - An Overview of Standard and Multilevel Prediction Algorithms

While standard prediction algorithms have proven to be valuable analytical tools (e.g. Chen et al., 2018; Kemper et al., 2020), they often do not explicitly account for the hierarchical data structures commonly encountered in fields such as genomics, medical studies, and the social sciences (Ngufor et al., 2019). This limitation may lead to biased predictions and limit model performance. To address these challenges, researchers have developed multilevel extensions de-

signed to account for dependencies within the data. While these approaches were initially created for longitudinal outcomes or high-dimensional data, they also apply to hierarchical datasets (Capitaine et al., 2021; Lin and Luo, 2019; Ngufor et al., 2019; Speiser, 2021; Speiser et al., 2019, 2020).

To provide a theoretical foundation, this section begins by describing the fundamental concepts of machine learning for classification, distinguishing between supervised and unsupervised learning, before delving into specific algorithms. For each method, a brief theoretical introduction is provided, followed by its multilevel extension, if applicable. Algorithms without existing multilevel variants but still relevant to the reviewed studies are also included. Practical considerations for implementing these approaches, including software and packages, are discussed.

2.1.1 Machine Learning for Classification

Machine learning is a branch of artificial intelligence that focuses on developing algorithms capable of learning existing patterns in the data, often in the context of large complex datasets. Machine learning techniques are often broadly categorized into two primary approaches: *supervised* and *unsupervised learning*. In unsupervised learning, algorithms are *fit* to or *trained* on or *learn* from unlabeled data, where there is no predefined outcome variable Y. These methods are particularly useful for exploratory data analysis, revealing patterns or groupings among data points. For example, an unsupervised learning algorithm might identify clusters of students with similar study habits, enabling educators to tailor instructional strategies or recommend personalized course materials based on each group's learning preferences (Chen et al., 2009). In contrast, supervised learning involves training algorithms on labeled data, where an outcome variable Y

is available. These techniques aim to establish relationships between predictor variables X and outcomes Y, allowing for either explanatory modeling (to understand relationships between an outcome Y and a set of variables X), or for predictive modeling (to make predictions for future observations). Supervised learning can be further divided into two types of tasks: *classification*, where the outcome is categorical (e.g., pass/fail), and *regression*, where the outcomes is continuous (e.g., scores on an exam). The remainder of this section focuses on supervised learning methods for classification tasks, which were central to this research. Table 2.3 displays the standard algorithms and the corresponding multilevel adaptations, along with implementation details.

2.1.2 Standard Algorithms with Multilevel Extensions

2.1.2.1 Logistic Regression

Logistic Regression is a classic parametric statistical algorithm widely used for binary classification. Employing a maximum likelihood estimation approach, it models the probability of an event occurrence (usually 0 or 1) based on predictor variables using the logistic function. The logistic function transforms a linear combination of the predictor variables into a probability estimate, which is expressed as

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}} \tag{2.1}$$

where $X = (X_1, \ldots, X_p)$ are p predictors. Parameters $\beta_0, \beta_1, \ldots, \beta_p$ can be estimated using Maximum Likelihood Estimation (MLE).

While logistic regression is known for its simplicity, interpretability, and ease of implementation, it relies on assumptions that can be challenging to validate in practice: 1) observations

should be independent of each other; 2) the relationship between the log odds of the dependent variable and the independent variables is linear; 3) the independent variables should not be correlated (i.e., no multicollinearity) (Gibson and Ifenthaler, 2017). Additionally, this method may face convergence issues and reduced performance with high-dimensional sparse data (Allison, 2008). Logistic Regression is a standard method and can be implemented in any statistical software package, for example by using the glm() function included in the `stats` package in R.

2.1.2.2 Generalized Linear Mixed Model

A generalized linear mixed model (GLMM) extends logistic regression by incorporating both fixed effects, which represent relationships assumed to be consistent across all clusters in the data, and random effects, which capture group-specific deviations from these overall relationships, making it suitable for multilevel data. The model for a GLMM with a random intercept and a random slope, P level-1 and Q level-2 predictors, X_p and Z_q respectively, takes the form

$$\begin{aligned}
\text{logit}(P(Y_{ij}=1)) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) &= \beta_{0j} + \sum_{p=1}^{P}\beta_{pj}X_{pij} \\
\beta_{0j} &= \gamma_{00} + \sum_{q=1}^{Q}\gamma_{0q}Z_{qj} + u_{0j} \\
\beta_{0j} &= \gamma_{10} + u_{1j} \\
\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma_{u01} \\ \sigma_{u01} & \sigma^2_{u1} \end{pmatrix}\right)
\end{aligned} \tag{2.2}$$

Model parameters are estimated using maximum likelihood or restricted maximum likelihood (REML) methods in frequentist approaches, or through maximum a posteriori estimation in Bayesian frameworks. Like Logistic Regression, GLMMs can face practical challenges, includ-

ing convergence issues and computational demands in high-dimensional data, as well as difficulties in validating parametric assumptions such as linearity and multicollinearity (Muschelli et al., 2014). GLMMs are widely supported in standard statistical software. In R, the `glmer()` function from the `lme4` package is commonly used for frequentist estimation, while the `blmer()` function from the `blme` package introduces Bayesian regularization with weakly informative priors. Alternatively, a full Bayesian approach can be implemented using the `MCMCglmm` package (Hadfield, 2010), which employs Markov Chain Monte Carlo (MCMC) techniques to estimate posterior distributions for model parameters.

2.1.2.3 Regularization Techniques: Lasso, Ridge, and Elastic Net

Regularization techniques are used for controlling model complexity, particularly in high-dimensional data contexts where overfitting is a concern. Lasso, ridge regression, and elastic net are popular regularization methods, each applying a different penalty function to balance complexity and predictive accuracy. For instance, Ridge regression penalizes large coefficients by adding an L2 penalty, which is the sum of the squared values of the coefficients. This technique shrinks coefficients towards zero but does not set them to zero, retaining all predictors in the model (Hoerl and Kennard, 1970). Ridge regression is particularly effective when dealing with multicollinearity, as it reduces variance without eliminating variables. Lasso, or Least Absolute Shrinkage and Selection Operator, adds an L1 penalty to the regression model, which is the sum of the absolute values of the coefficients (Tibshirani, 1996). Unlike ridge regression, lasso has the unique ability to set some coefficients exactly to zero, effectively performing variable selection. This property makes lasso particularly useful for high-dimensional data where interpretability

and parsimony are essential. Elastic net combines the L1 and L2 penalties of lasso and ridge regression to balance variable selection and model stability (Zou and Hastie, 2005). This method is advantageous when the dataset has many correlated predictors, as it can retain grouped variables and prevent some coefficients from being shrunk to zero too quickly.

The coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ are estimated by minimizing the quantity:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right) \tag{2.3}$$

The mixing parameter, α, balances the contributions of the L1 (lasso) and L2 (ridge) penalties. When $\alpha = 1$, Equation 2.3 simplifies to the lasso solution. When $\alpha = 0$, Equation 2.3 simplifies to the ridge solution. For values $0 < \alpha < 1$, Equation 2.3 represents the elastic net solution, combining aspects of both lasso and ridge regression.

The regularization parameter, λ, controls the strength of the penalty. When $\lambda = 0$, the model reduces to the ordinary least squares solution as no penalty is applied. As λ increases, the penalty strengthens which results in the shrinkage of coefficients. In extreme cases, this can lead to a null model where all coefficient estimates are zero, depending on the choice of the penalty (e.g., L1, L2, or a combination). In practice, the optimal value of λ is often selected through cross-validation to balance model fit and regularization.

In R, the `glmnet` package is commonly used for implementing these methods. The function `cv.glmnet()` facilitates cross-validated tuning of λ, and the resulting model can be used for making predictions.

2.1.2.4 Multilevel Lasso

The multilevel lasso algorithm, GLMMLasso, extends the lasso method to multilevel settings (Schelldorfer et al., 2014). It uses a similar L1-penalty as described in equation (2.3); however, it incorporates additional terms to accommodate the variance components of a multilevel model, as outlined in equation (2.2). GLMMLasso produces models that include a subset of the predictor variables while accounting for the nested structure of multilevel data. Although originally developed as a variable screening method to handle scenarios where the number of variables exceeds the sample size, GLMMLasso can also be applied for prediction purposes (Li et al., 2022; Wu et al., 2021). Researchers interested in fitting a multilevel lasso can use the R package `glmmLasso`.

2.1.2.5 Decision Trees

Decision trees, also called classification and regression trees (CART), are nonparametric machine learning methods that offer an intuitive approach to making predictions. Their output is often easy to interpret, as the tree structure provides a visual representation of the decision-making process, showing how predictions are made step by step (Hastie et al., 2009). Decision trees are considered nonparametric models because they don't make explicit assumptions about the functional form of the relationship between predictors and the outcome. Unlike parametric models that assume a predefined structure for the relationship between variables, decision trees are flexible and can capture nonlinear relationships between variables.

The algorithm classifies observations by creating splits on predictor variables by systematically searching through each predictor variable to determine which split for each variable would

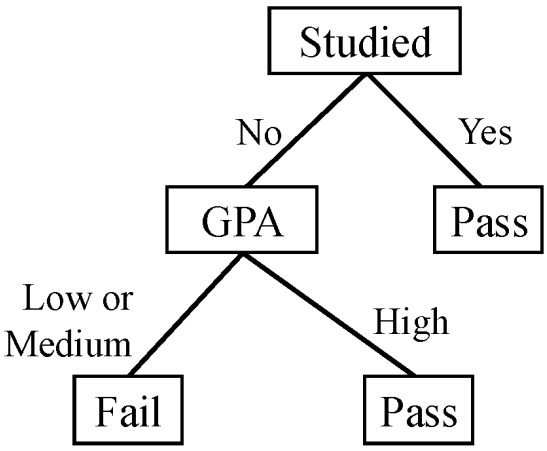

Figure 2.1: An example decision tree used to predict if a student will pass or fail an exam.

yield the most homogeneous subgroups with respect to the outcome variable. The variable leading to the most homogenous subgroups is selected, and the process iterates with the subgroups generated from the split. To prevent overfitting, a complexity parameter controls the number of splits in the tree and is typically selected via cross-validation. The result is a decision tree with rules that are used to classify an observation. In Figure 2.1, an example decision tree predicts whether a student will pass or fail an exam based on two predictor variables: whether they studied and their GPA.

While decision trees are an easily interpretable method, they often exhibit high variance, meaning that small changes in the training data can result in significantly different final trees. This sensitivity can impact their generalizability and performance compared to other machine learning methods (James et al., 2013). Several R packages, such as the `rpart` package, are

available for fitting decision trees.

2.1.2.6 Multilevel Decision Trees

Multilevel decision trees extend traditional decision trees to accommodate the hierarchical structure of multilevel data. Several extensions have been proposed including generalized mixed-effects regression tree (GMERT) Hajjem et al. (2017), generalized linear mixed-effects model tree (GLMM tree) Fokkema et al. (2018), multilevel classification and regression trees (M-CART) Lin and Luo (2019), binary mixed model tree (BiMM tree) Speiser et al. (2020), and generalized mixed effects tree (GMET) Fontana et al. (2021). These methods share the common theme of combining decision tree structures with multilevel models to handle multilevel data. Typically, they estimate fixed effects using a decision tree, and they estimate random effects using a multilevel model. While the overarching objective is consistent, these methods differ in estimation techniques (e.g., penalized quasi-likelihood (PQL), adaptive Gauss-Hermite quadrature (AGHQ), or Bayesian GLMM), computational strategies (iterative vs. non-iterative), and the specific decision tree and multilevel model used. A short summary of each method can be found in Table 2.1, and each method is detailed below.

Hajjem et al. (2017) first introduced GMERT, an extension of their original mixed effects regression tree (MERT) algorithm (Hajjem et al., 2011). GMERT replaces the linear modeling of fixed effects in GLMM with a decision tree, enabling it to capture complex, nonlinear relationships. The algorithm adapts the penalized quasi-likelihood (PQL) approach for GLMMs estimation, using a first-order Taylor-series expansion to create a weighted pseudo-model. GMERT iterates between updating the decision tree and the GLMM, incorporating the estimated random

intercept into the outcome variable at each step. Convergence is achieved when changes in likelihood values fall below a predefined threshold. Supplementary materials, including R code, are available in Hajjem et al. (2017).

The GLMM tree algorithm proposed by Fokkema et al. (2018) builds on model-based recursive partitioning generalized linear model (GLM) trees. Model-based recursive partitioning GLM trees are a type of decision tree that is constructed in a way that each node corresponds to a subgroup specific GLM. This means that instead of simply making decisions based on the predictor variables, each node of the tree also corresponds to a GLM that models the relationship between the variables and the outcome variable within that specific subgroup. This approach makes them suitable for complex, nonlinear relationships. GLMM tree algorithm extends the GLM tree by incorporating random effects into the model. The algorithm alternates between two steps: (1) assuming random effects are known and estimating fixed effects using a GLM tree, and (2) assuming the tree structure is fixed and estimating random effects using a GLMM. This iterative process continues until a convergence, typically when the tree does not change from one iteration to the next. This method can be implemented in R using the `glmertree` package.

Lin and Luo (2019) developed Multilevel Classification and Regression Trees (M-CART), which is very similar to GMERT. However, while GMERT uses the penalized quasi-likelihood (PQL) method for estimation, M-Cart uses the adaptive Gauss-Hermite approximation for parameter estimation. The algorithm decomposes the outcome into fixed and random components, using a standard decision tree to model fixed effects and a GLMM to model random effects. It iteratively updates these components, with each iteration refining the fixed effects based on updated random effects. Convergence is achieved when changes in log-likelihood values fall below a preset threshold. While no R package currently exists for M-CART, the authors provide

implementation details in the supplementary materials.

BiMM tree, introduced by Speiser et al. (2020), combines decision trees with Bayesian GLMMs for modeling clustered and longitudinal binary outcomes. The algorithm alternates between constructing classification and regression tree (CART) models and incorporating their outputs into a Bayesian GLMM framework to adjust for clustering. Supplementary materials, including R code, are provided in the original paper.

Fontana et al. (2021) introduced the generalized mixed effects tree (GMET) algorithm as an extension of the random effects expectation-maximization (RE-EM) tree (Sela and Simonoff, 2012), designed to handle a wider range of outcome types in the exponential family, including binary outcomes. Unlike the GLMM tree, which uses recursive partitioning, GMET uses a standard decision tree. A key distinction of GMET is that it is non-iterative, fitting the decision tree and GLMM components only once. This non-iterative approach significantly reduces computational time while retaining flexibility for modeling multilevel data. Although no dedicated R package exists for GMET, the authors provide R code in their supplementary materials.

Table 2.1: Summary of multilevel decision trees

Algorithm	Decision Tree Method	Mixed Effects Method	Iterative
GMERT	Standard Decision Tree	GLMM with PQL	Yes
GLMM tree	GLM tree	GLMM with AGHQ	Yes
M-CART	Standard Decision Tree	GLMM with AGHQ	Yes
BiMM tree	Standard Decision Tree	BGLMM	Yes
GMET	Standard Decision Tree	GLMM with AGHQ	No

2.1.2.7 Ensemble Methods: Boosting, XGBoost, and Random Forests

To address the high variance and overfitting challenge of decision trees discussed above, ensemble methods such as boosting, XGBoost and random forests may be used. Ensemble methods combine the predictions of multiple individual models to produce a more robust (i.e., less sensitive to small changes in the training dataset) and accurate overall prediction (Hastie et al., 2009). The underlying idea is that by aggregating the predictions of multiple models, the strengths of individual models can be emphasized, and their weaknesses can be mitigated. In the case of decision trees, ensemble methods build a collection of trees, each trained on different subsets of the data or with variations in the training process. By combining the predictions of these trees, ensemble methods provide a more generalizable and reliable prediction, reducing the risk of overfitting and enhancing performance on new, unseen data. A generic ensemble method is shown in Figure 2.2.

Boosting methods iteratively construct a sequence of trees, with each subsequent tree focused on the minimizing prediction error made by its predecessor. The process begins by fitting an initial tree to the data and calculating its prediction errors. In subsequent iterations, a new tree is trained to correct the residual errors of the previous model. This iterative process assigns greater weight to observations that were misclassified or poorly predicted in earlier iterations which focuses the model's attention on harder-to-predict cases. The final prediction is a weighted sum of the predictions from all individual trees in the sequence. The weights assigned to each tree are based on their effectiveness in reducing errors; trees that contribute more to the model's accuracy receive higher weights. The `gbm` package can be used in R for fitting boosted decision trees. XGBoost, or Extreme Gradient Boosting, is an advanced form of boosting known

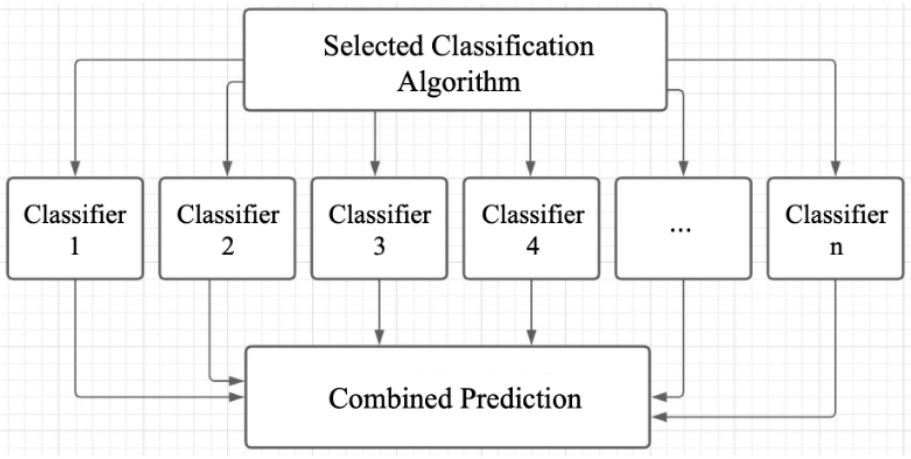

Figure 2.2: A generic ensemble method. Adapted from "A Comparative Performance Assessment of Optimized Multilevel Ensemble Learning Model with Existing Classifier Models," by Mukesh Kumar, Karan Bajaj, Bhisham Sharma, and Sushil Narang.A, 2022, *Big Data*. Copyright 2022 by Mary Ann Liebert, Inc., publishers.

for its computational efficiency and regularization capabilities (Chen and Guestrin, 2016). The `xgboost` package in R is widely used for implementing this powerful method.

Unlike boosting, which builds trees sequentially, random forests construct an ensemble of independent trees trained in parallel. The method relies on two key principles: bootstrap sampling and random feature selection. Bootstrap sampling involves creating multiple bootstrap datasets by randomly selecting observations with replacement from the original dataset. Each bootstrap dataset serves as a unique training set for a different tree in the forest. Additionally, during tree construction, only a random subset of predictors is considered for splitting at each node, further introducing diversity among the trees. This randomization enhances the model's robustness to minor changes in the training data which improves its prediction accuracy and increases its ability to generalize to new data. For classification tasks, the final prediction is determined through a majority vote, where each tree casts its vote, and the class predicted by the

majority becomes the final prediction. Random forest hyperparameters include the number of trees in the forest, the number of predictors considered for each split, and the depth of individual trees. The `randomForest` package in R is widely used for implementing random forests.

2.1.2.8 Multilevel Ensemble Methods

Several extensions of ensemble methods, particularly random forests and gradient boosting machines, have been proposed to accommodate multilevel data structures. These methods integrate ensemble approaches with multilevel models to account for the hierarchical nature of the data while leveraging the predictive power of ensemble techniques. Although they share the common goal of incorporating fixed and random effects, they differ in their implementation approaches, such as using Bayesian GLMMs (BiMM Forest) (Speiser, 2021) or frequentist GLMMs (MErf, MEgbm, and GMERF) (Ngufor et al., 2019; Pellagatti et al., 2021). An overview of these methods is provided in Table 2.2, with further details below.

BiMM Forest uses a two-step iterative process: (1) it fits a standard random forest model while treating the cluster effect as a known random effect, and (2) a Bayesian GLMM is fit assuming the fixed effects are known. Through this process, the algorithm refines it's predictions and combines the strengths of random forests and Bayesian GLMMs. An added feature is the ability to perform variable selection within the BiMM Forest framework, as introduced by Speiser (2021). Implementation guidance and R code are available in Speiser et al. (2019).

MErf and MEgbm were introduced as part of the mixed-effect machine learning (MEml) framework by Ngufor et al. (2019). MErf integrates random forests into a multilevel framework using an iterative algorithm to alternately estimate fixed effects via random forests and random

effects via GLMMs. MEgbm applies a similar iterative approach to Gradient Boosting Machines, where each tree corrects residual errors from previous iterations. These algorithms used to be available via the `Vira` package Mangino et al. (2022); Mangino and Finch (2021) but it is no longer available. The MEml GitHub repository is also no longer maintained.

GMERF, inspired by the GMET algorithm, replaces decision trees with random forests in the fixed-effects component of the model. This method separately estimates fixed and random effects and alternates between them until convergence is achieved. The details of the GMERF algorithm are provided in Pellagatti et al. (2021), though no R package is currently available.

Table 2.2: Summary of multilevel ensemble methods

Algorithm	Ensemble Method	Random Effects Estimation
BiMM Forest	Random Forest	Bayesian GLMM
MErf	Random Forest	GLMM
MEgbm	Gradient Boosting	GLMM
GMERF	Random Forest	GLMM

2.1.2.9 Neural Networks

Neural networks draw inspiration from the basic functioning of the human brain and how it interprets information. They are advanced models composed of interconnected nodes which are organized into layers: an input layer, one or more hidden layers, and an output layer (Goodfellow et al., 2016). These nodes resemble neurons in the brain, and the connections between them, which are represented by weights, carry information about the input signal, enabling the network to discern intricate patterns and relationships within the data. The process is as follows: first, the input layer receives data and connections with weights transmit this information to the hidden layers. These hidden layers, akin to brain synapses, perform computations using acti-

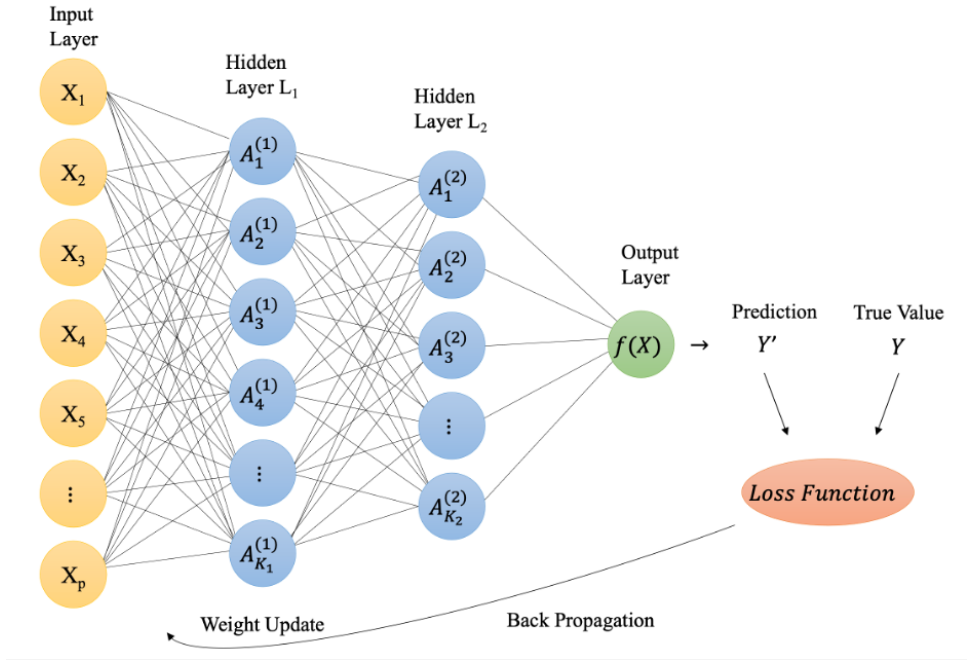

Figure 2.3: The learning process of a neural network with two hidden layers, L_1 and L_2, with activations $A_k = h_k(X)$. Here the activations A_k are not directly observed and the functions $h_k(X)$ are not fixed in advance but are learned during the training of the network. They represent nonlinear transformations of linear combinations of the input variables $X_1, X_2, \ldots, X_p$. The output layer is a linear model that uses these activations A_k as inputs, resulting in a function $f(X)$. Weights are updated through an iterative process until a loss function is minimized. Adapted from

vation functions that introduce nonlinearity to the model. Then, the output layer generates the final predictions. During training, the network adjusts its weights based on the errors between its predictions and the actual outcomes through a process called backpropagation. This iterative learning process allows neural networks to adapt and improve their ability to recognize patterns and make accurate predictions. Figure 2.3 provides a visual representation of a neural network with two hidden layers.

In essence, neural networks function as complex information processors, where each layer extracts increasingly abstract features from the input data. The iterative adjustments made on the

weighted connections between nodes enable the network to learn and adapt to the intricacies of the underlying patterns, making it proficient at capturing complex relationships within the data and allowing them to handle classification tasks that challenge traditional linear classifiers. However, their effectiveness comes with some considerations. Training large neural networks can be computationally intensive and time-consuming, potentially limiting their feasibility in resource-constrained environments (Thompson et al., 2023). Additionally, their "black box" nature, where the internal workings are not easily interpretable, has raised concerns about the lack of transparency and interpretability in the decision-making process (Guidotti et al., 2018). In R, neural networks can be implemented using various packages such as `keras` and `nnet`.

2.1.2.10 Panel Neural Networks

Panel Neural Networks (PNNs) were developed to extend neural networks for longitudinal (panel) data (Crane-Droesch, 2017). PNNs explicitly incorporate random effects into the network's architecture, capturing both within- and between-group variability. These random effects are jointly estimated alongside fixed effects, enabling the network to optimize predictions at multiple levels. During training, backpropagation is used to compute gradients for all parameters, ensuring that both global patterns and group-specific deviations are learned. `panelNNET` can be used to fit Panel Neural Networks.

2.1.3 Standard Algorithms without Multilevel Extensions

2.1.3.1 Naive Bayes

Naive Bayes is a simple yet effective algorithm for classification that relies on Bayes' theorem. The "naive" aspect comes from its assumption that predictors are independent of one another, which is often not the case in real-world data. Despite this, the algorithm frequently performs well and is easy to implement (Russell and Norvig, 2016). The algorithm calculates the probability of an observation belonging to a class C_k based on the predictors X, using the Bayes' Theorem:

$$P(C_k \mid X) = \frac{P(X \mid C_k) \cdot P(C_k)}{P(X)} \tag{2.4}$$

For example, Naive Bayes can be used to predict whether a patient has the flu based on symptoms such as chills, a runny nose, and fever. Using historical data, conditional probabilities of these symptoms given each class (e.g., flu or no flu) are computed. The Naive Bayes classifier then combines these probabilities and assigns the patient to the class with the highest probability. In this context, if the computed probabilities indicate a higher likelihood of the patient having the flu given their symptoms, the model predicts "flu." This method is particularly useful for quick and interpretable predictions. The `naivebayes` package in R provides tools for implementing this algorithm.

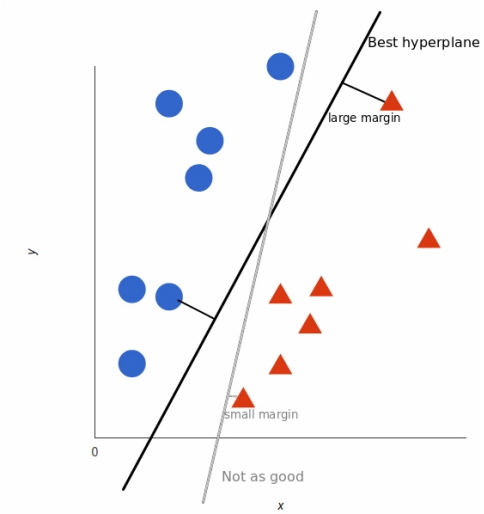

Figure 2.4: An illustration of two possible hyperplanes that separate the two classes. Reprinted from *Support Vector Machines (SVM) Algorithm Explained*, by B. Stecanella, June 22, 2017, https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/.

2.1.3.2 Support Vector Machines

Support vector machines (SVM) are versatile prediction algorithms for both linear and nonlinear classification tasks. SVM aims to find an optimal hyperplane (a multidimensional surface) that separates the different classes while maximizing the margin (the space between the hyperplane and the nearest data points of each class). Figure 2.4 displays a simple example with two hyperplanes, the optimal hyperplane is the one that creates the largest separation between the two classes.

SVM's performance depends on its tuning parameters, such as the regularization parameter C. This parameter plays a crucial role in balancing the trade-off between margin width and minimizing classification errors. The regularization parameter allows practitioners to control the penalty imposed for misclassification, thereby influencing the model's sensitivity to individual

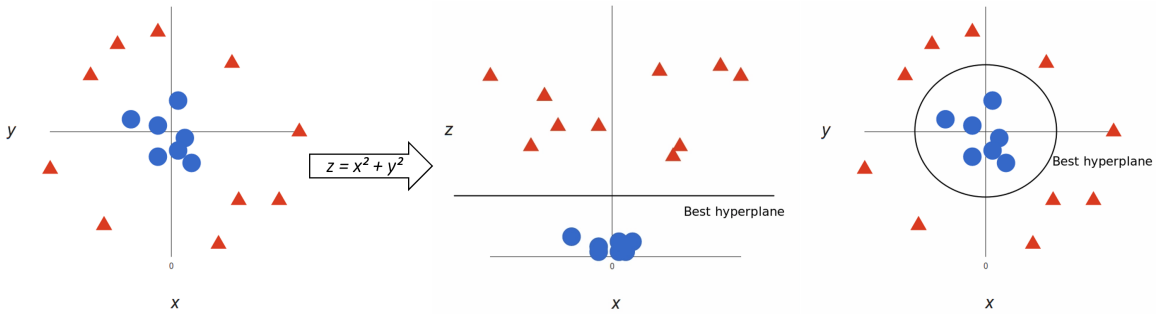

Figure 2.5: Using a nonlinear kernel to separate the two classes. Adapted from *Support Vector Machines (SVM) Algorithm Explained*, by B. Stecanella, June 22, 2017, https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/.

data points. In situations where classes exhibit nonlinear relationships and aren't easily separable by a straight line, SVM employs kernel functions to map the predictor variables into a higher-dimensional space. This transformation enables SVM to discern complex decision boundaries, accommodating intricate relationships within the data. The concept of kernel functions is akin to using a lens to view data patterns in a higher-dimensional realm, providing SVM with the capability to address the subtleties of nonlinear data structures effectively. Figure 2.5 provides a visual representation of how a kernel function can be used to classify nonlinear data by creating a new z dimension, where $z = x^2 + y^2$. The `e1071` package in R provides tools for implementing SVMs.

2.1.3.3 Stacking

Stacking is another powerful ensemble method that combines the predictions of multiple base models using a meta-learner, a model that learns how to best combine the outputs of the base models to make the final prediction. Unlike boosting and random forests, which typically

use decision trees, stacking can leverage a variety of different base models, such as decision trees, support vector machines, and linear models. Each base model is trained on the training data, and their predictions are then used as inputs for the meta-learner. This approach takes advantage of the unique strengths of each base model while mitigating their individual weaknesses, leading to a more robust and accurate model (Wolpert, 1992). In R, the `caretEnsemble` and `SuperLearner` packages can be used to implement stacking.

2.1.3.4 K-Nearest Neighbors (KNN)

K-nearest neighbors (KNN) is a simple, non-parametric machine learning algorithm that identifies the k closest training observations in the feature space to a given observation. It then assigns the most frequent class among these k neighbors as the predicted class for that observation. The algorithm's strength lies in its simplicity and adaptability to nonlinear relationships, making it particularly suitable for datasets where relationships between predictors and outcomes are not easily captured by linear models. In R, KNN can be implemented using several packages, with the `class` package being one of the most commonly used.

2.1.4 Balancing Flexibility and Interpretability in Model Selection

Another consideration when selecting an appropriate method is the balance between flexibility and interpretability. Flexibility allows models to capture complex patterns and nonlinear relationships, which can enhance predictive performance for complex datasets. However, highly flexible models often come at the cost of interpretability, earning them the designation of "black box" models. This lack of transparency can make it challenging for researchers and policymakers

Table 2.3: Standard and Multilevel Classification Algorithms

Standard Algorithm	**Multilevel Extensions**	**R Implementation (Standard)**	**R Implementation (Multilevel)**
Logistic regression	Generalized Linear Mixed Model (GLMM) Bayesian Generalized Linear Mixed Model	`stats`	`lme4` `blme`, `MCMCglmm`
Lasso	Multilevel Lasso (GLMMLasso)	`glmnet`	`glmmLasso`
Decision trees	Multilevel Classification and Regression Trees (M-CART) Generalized Mixed-Effects Trees (GMET) Generalized Linear Mixed-Effects Model Tree (GLMM tree) Generalized Mixed Effects Regression Tree (GMERT) The Binary Mixed Model Tree (BiMM tree)	`rpart`	Supplementary R code in (Lin and Luo, 2019) Supplementary R code in (Fontana et al., 2021) `glmertree` Supplementary R code in (Hajjem et al., 2017) Supplementary R code in (Speiser et al., 2020)
Gradient boosting machines	Mixed-Effects Gradient Boosting Machines (MEgbm)	`xgboost`, `gbm`	N/A; Deprecated: `Vira` package
Random forests	Mixed-Effects Random Forests (MErf) The Binary Mixed Model Forest (BiMM forest) Generalized Mixed-Effects Random Forest (GMERF)	`randomForest`, `ranger`	N/A; Deprecated: `Vira` package Supplementary R code in (Speiser et al., 2019) No R package; detailed in (Pellagatti et al., 2021)
Artificial Neural Networks	Panel Neural Networks (PNNs)	`keras`, `nnet`	`panelNNET`
Naive Bayes	None available	`naivebayes`	N/A
Support Vector Machines	None available	`e1071`, `caret`	N/A
Stacking	None available	`caretEnsemble`, `SuperLearner`	N/A
K-Nearest Neighbors	None available	`class`	N/A

to derive actionable insights or communicate findings effectively. Figure 2.6 provides a visual framework to help contextualize this trade-off. It maps the methods discussed in this section along two critical dimensions: flexibility and interpretability. Models like Logistic Regression and lasso are positioned in the high-interpretability/low-flexibility quadrant, reflecting their intuitive structures and ease of use, which are particularly valuable in applications where model transparency is important. On the other hand, algorithms such as Neural Networks and Gradient Boosting Machines, known for their adaptability and ability to handle complex interactions, occupy the high-flexibility/low-interpretability space. These models often offer improved predictive performance but may require additional techniques, such as feature importance analysis, to improve their explainability. This flexibility-interpretability trade-off suggests that the choice of model should align with the specific objectives and constraints of the research or policy analysis. For instance, policymakers aiming to identify broad trends and communicate findings to diverse stakeholders might prioritize more interpretable models. In contrast, researchers conducting exploratory analyses or predicting outcomes in complex data settings may opt for more flexible methods, even if interpretability is reduced.

2.1.5 An Overview of Evaluation Metrics for Classification

The evaluation of model performance often involves the division of data into training and testing sets. The training set is used to fit or train the model, allowing it to learn patterns and relationships in the data and adjust its parameters to make accurate predictions. The testing set, withheld during training, is used to evaluate the model's ability to generalize and make accurate predictions on new, unseen data. This process ensures that the model doesn't simply memorize

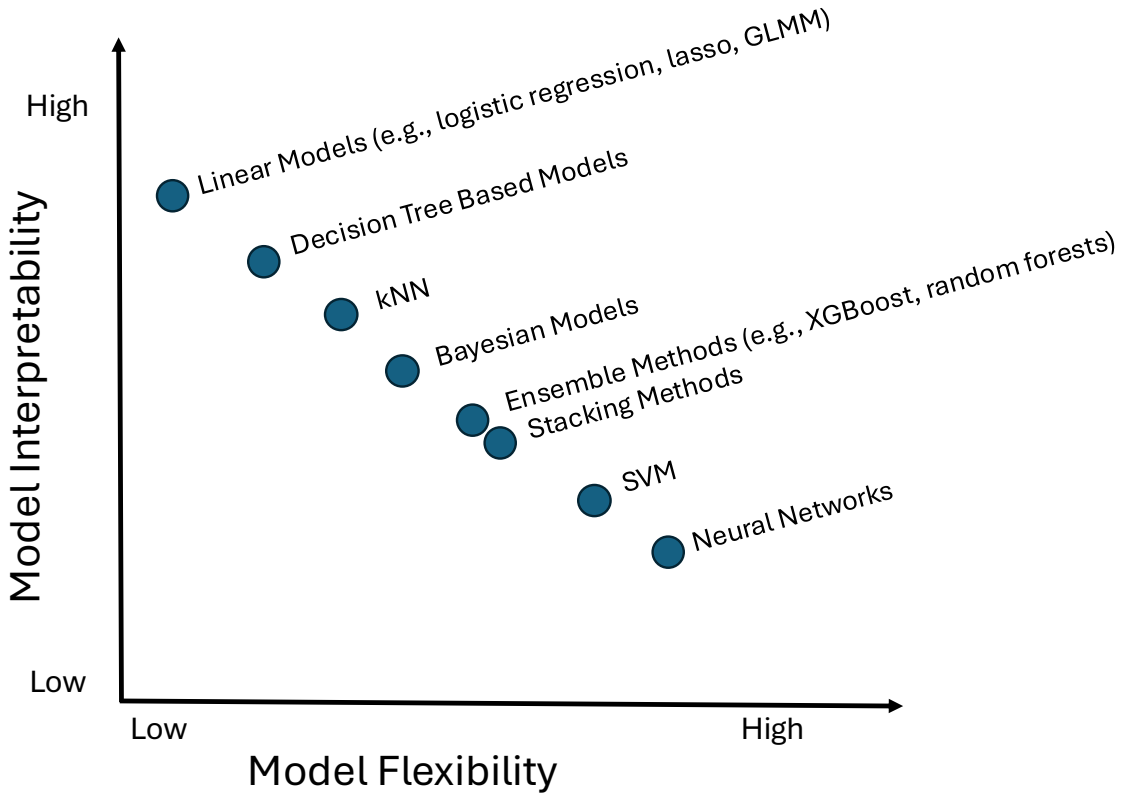

Figure 2.6: Illustration of the trade-off between model flexibility and interpretability. Adapted from *An introduction to statistical learning: With Applications in R (2nd ed.)* (p. 25) by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, 2021, Springer.

the training data but instead learns underlying patterns applicable to future observations. By comparing the model's predictions on the testing set with the true outcomes, a range of evaluation metrics can be calculated to assess its performance. This section introduces commonly used evaluation metrics, including classification accuracy, sensitivity, specificity, AUC-ROC, F1 score, Brier Score, and Predictive Mean Absolute Deviation (PMAD). These metrics provide insights into various aspects of model performance, which can guide decision-making in educational research.

Many of the metrics discussed here are derived from the confusion matrix, which summarizes true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. This matrix allows for the computation of key metrics and provides a clear visual representation of the model's performance across different outcomes. The confusion matrix layout is shown in Figure 2.7.

2.1.5.1 Classification Accuracy

Classification accuracy is the proportion of correct predictions made by the model. It is calculated as the sum of true positives and true negatives divided by the total number of cases. This straightforward measure provides a high-level view of overall model performance but does not reveal details about the distribution of errors among different classes. In educational research, this metric may be useful for quick comparisons between models, but it can be misleading if the dataset is imbalanced (He and Garcia, 2009). For example, when predicting dropout rates or other rare outcomes, reliance on classification accuracy alone may overestimate a model's effectiveness.

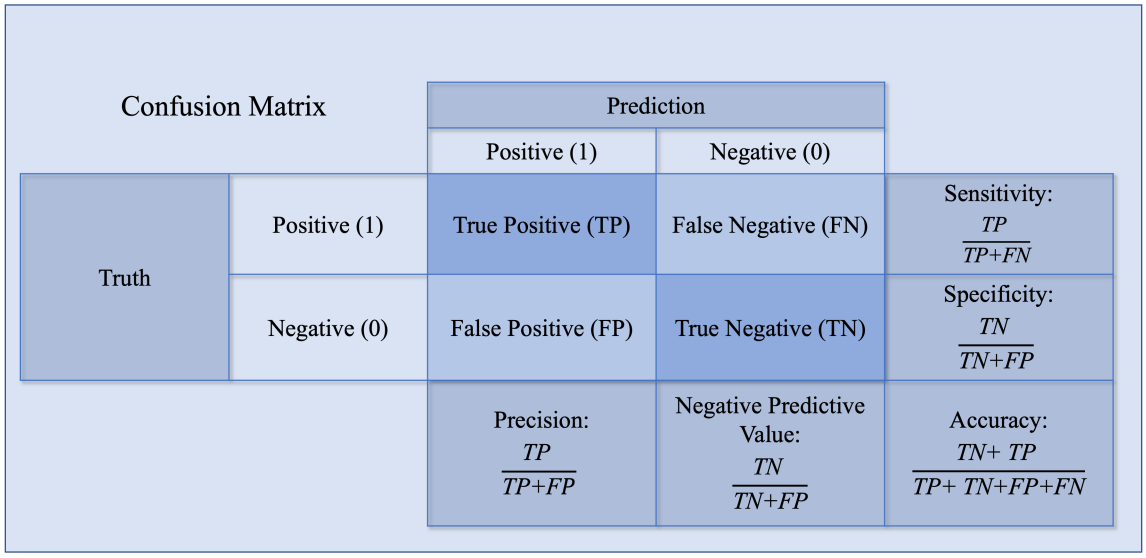

Figure 2.7: Formulation of common classification metrics based on the confusion matrix.

2.1.5.2 Sensitivity (True Positive Rate or Recall)

Sensitivity, or recall, measures the model's ability to correctly identify positive cases. It is defined as the ratio of true positives to the sum of true positives and false negatives:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.5}$$

High sensitivity indicates that the model effectively captures most positive cases. This metric is crucial in educational contexts where failing to identify true positive cases, such as students at risk of academic failure or needing special intervention, could have significant consequences (Powers, 2020). Sensitivity ensures that at-risk students are flagged, even if it means some false positives are also identified.

2.1.5.3 Specificity (True Negative Rate)

Specificity measures the model's ability to correctly identify negative cases. It is calculated as the ratio of true negatives to the sum of true negatives and false positives:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{2.6}$$

Specificity is important when the cost of false positives is high. For example, in educational screening programs where resources are limited, identifying students who do not need extra support helps avoid unnecessary allocation of interventions. A high specificity value ensures that fewer students are incorrectly flagged as needing support.

2.1.5.4 AUC (Area Under the ROC Curve)

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) evaluates the model's ability to distinguish between positive and negative cases across various threshold values. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at different thresholds. The AUC quantifies the area under this curve, ranging from 0 to 1, with higher values indicating better discrimination between classes, where 1.0 corresponds to perfect classification and 0.5 corresponds to performance equivalent to random guessing. The AUC-ROC is useful when comparing models because it reflects the trade-off between sensitivity and specificity (Huang and Ling, 2005). In education research, this metric can be valuable for evaluating predictive models where both correctly identifying at-risk students and minimizing false positives are important.

2.1.5.5 F1 Score

The F1 score is a balanced metric that combines both precision and recall (sensitivity); both metrics are detailed in Figure 2.7. It is calculated as the harmonic mean of these two metrics, providing a single metric that considers both false positives and false negatives:

$$\text{F} - 1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.7}$$

The F1 score is particularly valuable in educational research with imbalanced datasets, such as when only a small proportion of students are at risk. By weighing precision and recall equally, the F1 score helps assess models where capturing true positives is as important as minimizing false positives.

2.1.5.6 Brier Score

The Brier Score measures the mean squared difference between the predicted probabilities and the actual outcomes:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} (p_i - y_i)^2 \tag{2.8}$$

where, p_i represents the predicted probability for individual i, and y_i is the actual outcome for individual i (1 for positive, 0 for negative). A lower Brier Score indicates better model performance, as it implies the predicted probabilities are closer to the true outcomes. This metric is particularly valuable when predictions need to represent likelihoods rather than definitive classifications (BRIER, 1950). In educational settings, the Brier Score is useful when probabilistic predictions are needed, such as assessing the likelihood of students transitioning out of a STEM

field.

2.1.5.7 Predictive Mean Absolute Deviation (PMAD)

The Predictive Mean Absolute Deviation (PMAD calculates the mean absolute difference between the predicted probabilities and the actual outcomes:

$$\text{PMAD} = \frac{1}{N} \sum_{i=1}^{N} |p_i - y_i|. \tag{2.9}$$

This metric quantifies how close the predicted probabilities are to the actual outcomes on average. A lower PMAD value indicates better predictive accuracy, with values closer to zero suggesting a more accurate model. Like the Brier Score, PMAD relies on estimated probabilities, making it less suitable for methods that primarily generate class predictions, such as Support Vector Machines.

2.1.5.8 Summary

These evaluation metrics collectively offer insights into different aspects of model performance. Sensitivity and specificity are particularly useful when evaluating models in education research that involve trade-offs between identifying true positives and minimizing false positives. Metrics like the AUC-ROC, F1 score, Brier Score, and PMAD offer additional layers of insight, enabling researchers to choose the best model for their specific classification task.

2.2 Literature Review

The literature review provides context for this dissertation by examining existing applications of machine learning in education and previous comparisons of standard and multilevel methods.

2.2.1 Applications of Machine Learning Algorithms in Education Research

Machine learning has emerged as a powerful tool in education research, particularly for predicting student outcomes such as academic success, dropout risk, and behavioral issues. Despite the hierarchical nature of many educational datasets, where students are nested within classrooms, schools, or programs, researchers often default to standard (single-level) machine learning methods. A systematic review of machine learning applications in education highlights decision trees, logistic regression, neural networks, Naive Bayes, k-nearest neighbors, and support vector machines as the most frequently used prediction algorithms (Luan and Tsai, 2021; Sandra et al., 2021; Shahiri et al., 2015), as summarized in Table 2.4. This section reviews applications of machine learning in education, discusses the utility and limitations of standard approaches, and highlights the potential of multilevel prediction algorithms.

2.2.1.1 The Current Utility of Standard Machine Learning Methods

Standard machine learning algorithms have demonstrated substantial utility in predicting various educational outcomes. For example, Hutt et al. (2018) used random forests to predict whether students would graduate from college within four years, achieving 71.4% accuracy using a broad range of predictor variables, including sociodemographic data, institutional graduation

rates, academic achievements, standardized test scores, participation in extracurricular activities, work experience, and teachers and high-school guidance counselor ratings. Similarly, Gray et al. (2016) applied k-nearest neighbors to identify students at risk of failing their first year, achieving a 72% accuracy rate. Further evidence of the effectiveness of standard methods is provided by Kemper et al. (2020) who achieved 83% accuracy in predicting student dropout after one semester and 95% accuracy after three semesters using decision trees. These studies underscore the practical value of standard machine learning algorithms in identifying at-risk students and their potential for informing targeted interventions.

However, educational datasets often exhibit hierarchical structures, where students are nested within groups such as majors, institutions, or schools. Despite this, these group-level structures are frequently not integrated into predictive modeling frameworks. Instead, researchers often rely on standard machine learning methods that implicitly assume independence between observations, potentially overlooking contextual dependencies introduced by clustering. For example, Chen et al. (2018) predicted STEM major dropout using logistic regression, decision trees, random forests, and boosting methods. Their analysis revealed significant differences in how predictor variables influenced dropout rates across nine STEM majors. To address these differences, the study built nine separate predictive models, one for each major. While this approach captured group-specific trends, it did not explicitly model the shared or contextual influences of clustering. An alternative approach could involve including group identifiers as fixed effects in standard models. This would allow the model to account for group-level variability without requiring the complexity of a multilevel framework.

In another study, Huynh-Cam et al. (2021) used decision trees, random forests, and neural networks to predict first-year university performance and identify students who could benefit

from targeted interventions. The study consistently identified students' majors as among the most influential predictors, highlighting the importance of contextual effects. Interestingly, the researchers noted that the departments included in the study were randomly selected. If their goal was to generalize findings to a broader range of departments, a multilevel modeling approach could have been more appropriate. Multilevel models allow for the explicit modeling of random effects, enabling predictions that generalize beyond the specific clusters (e.g., departments) observed in the study.

In summary, standard machine learning algorithms have demonstrated substantial utility in educational research, providing effective tools for predicting outcomes such as student dropout and performance. These methods are often straightforward to implement and yield strong performance metrics in many applications. However, their reliance on the assumption of independence between observations may limit their ability to fully capture the hierarchical nature of educational data, where contextual dependencies introduced by clustering are common.

2.2.1.2 Analyses Incorporating Multilevel Prediction Algorithms

A smaller body of work have explicitly incorporated multilevel machine learning algorithms to address the limitations of standard methods. For example, Cannistrà et al. (2020) compared logistic regression, decision trees, random forests, GLMM, GMERT, and GMERF to predict students at risk of dropping out during their first year of college, where students were nested in degree programs with an ICC around 0.09. Their results showed that multilevel models like GMERF and GLMM consistently outperformed standard algorithms, achieving higher AUC and sensitivity scores by accounting for group-level variation. Similarly, Möller et al. (2022) demon-

strated the advantages of GLMM over logistic regression and decision trees for predicting school transition rates with students nested in classes which were nested in schools. All three methods were provided student-level variables which were also aggregated at the class and school level as additional predictors. Random intercept models and random intercept random slope models were considered. GLMM with a random intercept achieved the lowest error rates and Brier scores and highest AUC. However, both studies did not split their training and testing data at the cluster level, which may artificially inflate prediction performance. Mangino et al. (2021) also provided additional evidence of the potential benefits of multilevel approaches by applying mixed-effects random forests to predict student retention, which revealed slightly improved accuracy over standard methods like logistic regression and random forests.

2.2.1.3 Summary

Machine learning plays an important role in predicting educational outcomes, offering researchers and practitioners tools to identify at-risk students, optimize resource allocation, and design targeted interventions. While standard methods remain the default choice for many researchers, they may fall short in datasets with strong group-level dependencies. Multilevel prediction algorithms have demonstrated the potential for more accurate predictions, increased interpretability, and better generalizability when data are hierarchical by design. Additionally, due to the lack of clear guidance in the current literature, the potential consequences of not using multilevel algorithms for prediction remain unclear. This review showed that although standard predictive models have been successfully applied in education research, multilevel prediction methods are underutilized, presenting an opportunity for further exploration and improvement in

this field.

2.2.2 Comparisons of Standard and Multilevel Classification Algorithms on Hierarchical Data

While the number of multilevel prediction algorithms continues to grow, a notable gap in research persists in directly comparing the predictive performance of standard algorithms to their multilevel counterparts. The limited accessibility of multilevel prediction algorithms contributes to this gap, limiting studies primarily to the authors proposing these algorithms (e.g. Fokkema et al., 2018; Fontana et al., 2021; Hajjem et al., 2017; Ngufor et al., 2019; Pellagatti et al., 2021; Speiser et al., 2019, 2020) . Nevertheless, a growing body of independent research is beginning to address this gap through Monte Carlo simulations (e.g. Mangino, 2021; Mangino et al., 2022) and empirical analyses (e.g. Cannistrà et al., 2020; Möller et al., 2022; Philipp Kilham et al., 2019). This section highlights independent comparisons of these methods as well as studies introducing new approaches. Collectively these studies provide insights into the comparative strengths and weaknesses of standard and multilevel prediction algorithms in diverse contexts.

2.2.2.1 Comparative Performance of Standard and Multilevel Algorithms: Simulation Studies

One independent simulation study had a primary focus on comparing the performance of standard and multilevel algorithms. In this study, Mangino et al. (2022) found minimal differences in classification accuracy between standard and multilevel algorithms. The study compared

Table 2.4: Applications of machine learning algorithms in education research

	Algorithm												
Study	OLS	LR	DT	RF	GBM	KNN	NB	ANN	SVM	CPH	GLMM	M-DT	M-RF
Aguiar et al. (2014)		X	X	X			X						
Altabrawee et al. (2019)		X	X				X	X					
Bird et al. (2021)	X	X		X	X			X		X			
Cannistrà et al. (2020)		X	X	X							X	X	X
Chen et al. (2018)		X	X	X	X		X			X			
Jayaprakash et al. (2014)		X	X				X		X				
Gray et al. (2016)		X	X			X	X	X	X				
Hutt et al. (2018)		X	X	X	X		X						
Huynh-Cam et al. (2021)			X	X				X					
Kemper et al. (2020)		X	X										
Kotsiantis et al. (2003)		X	X			X	X	X					
KOTSIANTIS et al. (2004)		X	X			X	X	X	X				
Lee et al. (2021)								X					
Lee and Chung (2019)				X	X								
Mangino and Finch (2021)		X		X							X		X
Mayilvaganan and Kalpanadevi (2014)			X			X	X						
Möller et al. (2022)		X	X								X		
Elakia et al. (2014)			X										
Osmanbegović and Suljic (2012)			X				X	X					

logistic regression, GLMM, random forests, and MERF using simulated data generated using the sim.multi() function from the psych R package under several conditions commonly encountered in hierarchical data, such as sample size (number of clusters: 10, 50, 100; number of individuals per group: 10, 50, 100), ICC (0.1, 0.3, 0.8), class imbalance (50:50; 75:25; 90:10), and the level of predictors included in the data-generating model(3 at level-1; 2 at level-1 and 1 at level-2; 1 at level-1 and 2 at level-2). They also evaluated each algorithm's performance using the Program for International Student Assessment (PISA) dataset to predict which students would be held back in school based on several student-level predictors and two school-level predictors. Across the simulation conditions and in their empirical data analysis, they found limited evidence that the use of fixed versus mixed-effects models significantly affected prediction accuracy with respect to large-group recovery, small-group recovery, AUC, and binary cross-entropy. These findings suggest that simpler models like logistic regression or random forests might suffice for predictive classification. As a result, the authors suggest it may be unnecessary for researchers to use multilevel classifiers for predictive classification, as the results largely did not differ from the fixed-effects classifiers. Additionally, they did not find evidence that sample size affected prediction accuracy. However, they did find that by including both level-1 and level-2 variables, prediction accuracy could be improved. Therefore, they encouraged researchers to consider the type of predictors available and the underlying data structure, as these factors held more significance than the specific method chosen.

A second simulation study by Mangino and Finch (2021) compared five multilevel classifiers alongside one standard classifier. While the primary focus of their study was on the relative performance of different multilevel classifiers, with less emphasis on the comparison between standard and multilevel methods, they did include Naive Bayes in the set of classifiers under

examination. While the study demonstrated that the Multilevel Bayesian classifier and Panel Neural Networks outperformed other multilevel classifiers and Naive Bayes, it also highlighted Naive Bayes' notable robustness. Naive Bayes showed comparable performance and consistent accuracy across varying ICCs, sample sizes, and class imbalances, raising questions about whether the added complexity of more advanced multilevel models provides meaningful benefits. Its computational efficiency and stability suggest it as a viable option for hierarchical data. However, it is important to highlight that the study's exclusion of other standard classifiers, such as random forests or logistic regression, limits its ability to draw broader conclusions about the relative performance of standard and multilevel methods.

Across both of these simulation studies, the importance of predictor inclusion (both level-1 and level-2) and data structure emerged as critical factors influencing model performance. While multilevel models occasionally outperformed their standard counterparts, the minimal performance increase may not outweigh the additional complexity. Furthermore, both studies utilized the Vira package to implement multilevel random forests and gradient boosting machines (MERF and MEgbm, respectively); however, this package is no longer supported, presenting a significant barrier for researchers seeking to use these methods.

2.2.2.2 Comparative Performance of Standard and Multilevel Algorithms: Empirical Analyses

In addition to these independent simulation studies, three empirical analyses compare the relative performance of standard and multilevel algorithms. The empirical findings present a more nuanced view of the standard vs. multilevel debate, with results varying based on the application

context.

Kilham et al. (2019) examined the prediction of tree-level wood supply nested within forestry plots, providing a natural hierarchical structure for their analysis. This study compared random forests with generalized linear mixed models (GLMMs), focusing on eleven tree- and eleven plot-level predictors and tree-level outcomes, with training and testing sets split at the plot-level (between-group split). The study highlighted the flexibility of random forests in modeling nonlinear relationships and complex interactions, which achieved predictive accuracy and explained variance comparable to, or better than, GLMMs. Random forests' ability to adapt without requiring predefined interactions underscored its practical advantages, even in contexts with inherent multilevel structures introduced through multistage sampling. However, the study did not explicitly report ICC values, and therefore the results are limited in their generalizability which raises questions about random forests' ability to generalize in settings with potentially different hierarchical effects (such as higher ICC) than dataset.

In contrast, both Cannistrà et al. (2020) and Möller et al. (2022) found that multilevel machine learning algorithms offered improved prediction performance.

Cannistrà et al. (2020) investigated student dropout in Italian universities, using a range of standard models including logistic regression, decision trees, random forests, and multilevel algorithms such as GLMM, GMERT and GMERF. The study considered hierarchical data where students nested where within degree programs with an ICC ranging between 0.08 and 0.10. Predictors included both student-level variables (e.g., grades, demographics) and program-level factors, capturing interactions across levels. Multilevel methods, particularly GMERF and GLMM, consistently outperformed standard algorithms in terms of AUC and sensitivity. The study highlighted that incorporating both levels of predictors was essential for accurate dropout predictions

and demonstrated the superiority of multilevel approaches in contexts where hierarchical effects are pronounced. However, the authors also noted that standard models could perform well when guided by strong domain knowledge to carefully select predictors and model their relationships with the outcome variable.

Additionally, Möller et al. (2022) extended the discussion of multilevel methods for educational data, predicting school transition rates using standard decision trees, logistic regression and GLMM. The dataset was structured with students nested within schools, but unfortunately the study did not report specific ICC values. In this study, GLMM outperformed the other standard models, which indicates the importance of accounting for multilevel structures. However, they noted that trees had the advantage of handling missing data effectively, which traditional parametric models could not.

Table 2.5 below provides more details about study design and findings of the independent comparisons.

Table 2.5: Existing independent comparisons of standard and multilevel algorithms

Paper	Methods Compared		Study Design	Training: Testing Split	Metrics Reported	Findings
	Standard	Multilevel				
Cannistrà et al. (2020)	Decision Trees; Logistic Regression; Random Forest	GLMM; GMERF; GMERT	Empirical Analysis. Students nested within majors.	Observation level split. Training set: 5 years of cohorts; Testing set: single year cohort	AUC; Sensitivity	AUC and sensitivity, is always higher in multilevel models when predicting first-year student dropouts.
Philipp Kilham et al. (2019)	Decision Tree; Random forests	GLMM	Empirical Analysis. Trees nested within plots. Eleven plot-level and 11 tree-level variables were considered as predictors.	75:25 Cluster-level split: 5684 training set level-2 units with 49,055 level-1 units; 1894 test set level-2 units with 15,960 level-1 units	Variance Explained; Root Mean Squared Error	RF algorithms were generally superior to the GLMM with respect to both outcome metrics.

Mangino and Finch (2021)	Logistic Regression; Random Forest	GLMM; MERF	Simulation Study. Manipulated Factors: Sample Size; ICC; Class Imbalance; Level of predictor variables (level-1 vs level-2). 500 replications.	Cluster-level 50:50 Training: Testing Split	Specificity; Sensitivity; AUC; Binary Cross-entropy	Found limited evidence indicating any appreciable difference between fixed and mixed effects models on prediction accuracy or computational accuracy metrics. Empirical analysis revealed that regardless of the model selected, all of the prediction accuracy metrics were approximately equivalent
Mangino et al. (2022)	Naive Bayes Classifier	Multilevel Bayesian Classifier; MEGBM; MERF; Panel Neural Networks; REEMTree	Simulation Study. Manipulated Factors: Sample size ICC; Correlation between level-2 clusters; Class Imbalance. 100 replications.	Cross-validation	Overall Accuracy; Specificity; Sensitivity; Binary Cross-Entropy	Multilevel Bayes and Panel Neural Networks had the highest rates of classification accuracy and were robust to changing data conditions.
Möller et al. (2022)	Logistic Regression; Decision Tree	GLMM	Empirical Analysis.	Observation-level Split: 8,520 cases were randomly partitioned into a training set of size 5,000 and a test set of size 3,520. Repeated this 50 times.	AUC; Brier Score; Misclassification Rate;	GLMM outperformed logistic regression and decision tree.

2.2.2.3 Comparative Performance of Standard and Multilevel Algorithms: Studies Conducted by Proposing Authors

A significant portion of the literature comparing standard and multilevel algorithms originates from studies proposing new methods. While these studies may reflect potential conflicts of interest, such as selecting data conditions that emphasize the utility of their methods, they can still offer insights into the scenarios where multilevel algorithms outperform standard counterparts or, alternatively, when standard methods suffice. These studies highlight that multilevel algorithms can improve prediction in the presence of random effects, yield more stable results, and handle complex datasets effectively.

Multilevel methods are particularly advantageous when data exhibit random effects. For example, Speiser et al. (2019) found that BiMM Forest consistently achieved prediction accuracy equal to or better than standard random forest, GLMM, and BiMM Tree. Notably, BiMM Forest outperformed standard random forest when predicting new observations in clusters included in the training data but performed similarly when predicting observations in new clusters. These findings were corroborated by Speiser et al. (2020), who demonstrated that methods multilevel methods, such as BiMM Trees and Bayesian GLMM, yielded higher training set accuracy than standard decision trees, emphasizing the importance of accounting for hierarchical structures when data truly are hierarchical. Similarly, Hajjem et al. (2017) found that GMERT outperformed both standard decision trees and GLMM in the presence of random effects. Their findings suggested that ignoring random effects may lead to significant performance losses. Additionally, Fontana et al. (2021) showed that decision trees (e.g., GMERT, GLMM Tree, and GMET) outperformed standard decision trees when random effects were present but interestingly performed

worse than standard decision trees when random effects were absent.

Multilevel methods often provide more stable predictions across varying conditions. For instance, Pellagatti et al. (2021) found that GMERF produced stable predictions across different simulation settings, even when traditional parametric methods like logistic regression marginally outperformed GMERF in terms of accuracy. Similarly, Fontana et al. (2021) observed that GMET exhibited less variability and higher sensitivity compared to standard decision trees in real-world datasets, such as predicting student dropouts in hierarchical data.

Studies also emphasize the utility of multilevel machine learning methods for datasets with complex covariate structures. Pellagatti et al. (2021) highlighted that while parametric models like GLMM are effective with linear, uncorrelated covariates, multilevel machine learning methods like GMERF outperform them when covariates are numerous, nonlinear, or interactive. Ngufor et al. (2019) similarly demonstrated that the MEml framework effectively handled high dimensionality and nonlinear relationships, outperforming standard machine learning models for longitudinal and hierarchical data. Additionally, both BiMM forest and BiMM tree showed improved performance over Bayesian GLMM, standard decision trees, and random forests when predictors lacked a linear relationship with the outcome and when dealing with large random effects. Suggesting that not only were they better suited than standard algorithms but also served as potential alternatives to using parametric multilevel methods for complex datasets.

Although multilevel methods are often advantageous, there are scenarios where standard methods perform equally well or better. For instance, Fontana et al. (2021) found that standard decision trees outperformed multilevel classifiers when random effects were absent. Similarly, Speiser et al. (2019) observed that standard random forest performed as well as multilevel methods when predicting observations for clusters not represented in the training set. These findings

suggest that the choice of method should be guided by the presence or absence of hierarchical structures in the data.

Multilevel models often outperform standard methods when data exhibit hierarchical structures, large random effects, or nonlinear relationships between covariates and outcomes. However, standard methods may suffice, or even excel, when random effects are negligible or covariates exhibit simpler relationships. These findings underscore the importance of considering data characteristics, such as intraclass correlation, and fixed effects complexity, when selecting predictive algorithms.

2.2.3 Summary

In conclusion, the existing literature offers mixed findings regarding the performance of multilevel prediction algorithms compared to standard algorithms for classification tasks. A key takeaway from these studies is that selecting the most suitable algorithm is often context-dependent, influenced by factors such as level of ICC, dataset structure, and available predictor variable. The complexity of this decision and the lack of standardized guidelines highlight the need for further investigation to clarify when multilevel methods are necessary and when standard methods suffice. By comparing the performance of these algorithms across diverse contexts and datasets, this dissertation provides insights to better inform researchers' choices for predictive modeling in hierarchical data settings.

Chapter 3: Research Design

3.1 Simulation Study

This study used Monte Carlo simulation to compare the performance among several standard prediction algorithms and multilevel prediction algorithms under various manipulated conditions. All data was generated and analyzed using R 4.3.2 (R Core Team, 2016). The specifics of the simulation study design are outlined in the following section.

3.1.1 Manipulated Factors

The simulation study included four manipulated factors: the data-generating model, the amount of residual level-2 variance, sample size, and the way the training: testing datasets are split. This full factorial simulation design resulted in a total of 216 conditions (6 data-generating models * 3 residual level-2 variance * 6 sample sizes * 2 training: testing splits). Each set of conditions was replicated five hundred times. The selection of these conditions was informed by the current literature and addresses challenges encountered by researchers. The full set of conditions can be found in Table 3.1.

The first manipulated factor in the simulation study is the data-generating model. This factor is critical as it allows for the assessment of each algorithm's performance across diverse real-

Table 3.1: Manipulated factors and levels in simulation study

Manipulated Factors	Levels
Data Generating Model	Main Effects Random Intercept; Quadratic Nonlinear Random Intercept; Interaction Nonlinear Random Intercept; Exponential Nonlinear Random Intercept; Random Intercept and Random Slope; Random Intercept and Random Slope with Cross-Level Interaction
Residual Level-2 Variance	$\sigma^2_{u0} = 0.001; 2; 4$
Sample Size	
Number of clusters	$30; 50; 100$
Cluster Size	$50, 100,$
Training: Testing Split	Within Cluster Split; Cluster-Level Split

world data settings while capturing the linear and nonlinear patterns often present in educational datasets (Huang et al., 2010; McArdle and Ritschard, 2013). The selected models, including a random intercept model, three nonlinear random intercept models, a random intercept and random slope model, and a random intercept and random slope with a cross-level interaction model, collectively cover a spectrum of relationships between predictors and outcomes. The simulation also addressed the challenges associated with increasing dataset complexity. Traditional statistical methods often struggle to capture underlying patterns in such data, whereas machine learning methods are known for their ability to identify complex relationships and interactions without requiring explicit model specifications (Hastie et al., 2009; Janiesch et al., 2021). To evaluate the performance of standard and multilevel prediction algorithms under a range of realistic scenarios, the study incorporated six distinct two-level data-generating models, referred to as Models A through F (Table 3.2). These models were designed to reflect varying degrees of complexity and interactions commonly encountered in hierarchical datasets. Model A (Main Effects Random Intercept) represents a straightforward linear relationship between predictors and outcomes,

with variability captured at the cluster level through random intercepts. Building on this foundation, Model B (Quadratic Nonlinear Random Intercept) introduces nonlinearity by including quadratic terms for selected predictors to simulate datasets with curved relationships. Model C (Interaction Nonlinear Random Intercept) adds interaction terms among predictors to capture how combined effects can influence outcomes. Model D (Exponential Nonlinear Random Intercept) represents even greater complexity by applying exponential transformations to predictors, simulating nonlinear effects that may be challenging for traditional statistical methods to capture without strong theoretical beliefs about the underlying relationships between predictors and outcomes. Model E (Random Intercept and Random Slope) extends the random intercept model by introducing random slopes, allowing the relationship between a predictor and outcome to vary across clusters. Finally, Model F (Random Intercept and Random Slope with Cross-Level Interaction) includes cross-level interactions, where the relationship between individual-level predictors and the outcome is moderated by cluster-level variables. For instance, in Model F, the random slope β_{1j} was predicted by a level-2 variable Z_{1j}, reflecting scenarios where higher-level contexts influence individual-level outcomes. Each data-generating model shared common structural elements: $\mu_{ij} = \text{logit}^{-1}(\eta_{ij})$ and $y_{ij} \sim \text{Bern}(\mu_{ij})$, where y_{ij} represents the binary outcome for a level-1 unit (e.g., a student) i in level-2 unit (e.g., a classroom) j. The table below (Table 3.2) outlines the model-specific terms, highlighting how predictors, random effects, and interactions were incorporated in each condition.

Table 3.2: Data-generating models and corresponding equations

Model	Model Description	Model-Specific Terms
A	Main Effects Random Intercept	$\eta_{ij} = \beta_{0j} + \beta_1 X_{1ij} + \ldots + \beta_{15} X_{15ij}$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u0})$.
B	Quadratic Nonlinear Random Intercept	$\eta_{ij} = \beta_{0j} + \beta_1 X_{1ij} + \ldots + \beta_{15} X_{15ij} + \beta_{16} X^2_{1ij} + \beta_{17} X^2_{2ij}$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u0})$.
C	Interaction Nonlinear Random Intercept	$\eta_{ij} = \beta_{0j} + \beta_1 X_{1ij} + \ldots + \beta_{15} X_{15ij} + \beta_{16} X_{1ij} X_{2ij} + \beta_{17} X_{1ij} X_{3ij}$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u0})$.
D	Exponential Nonlinear Random Intercept	$\eta_{ij} = \beta_{0j} + \beta_1 X_{1ij} + \ldots + \beta_{15} X_{15ij} + \beta_{16} \exp(X_{1ij}) + \beta_{17} \exp(X_{2ij})$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u0})$.
E	Random Intercept and Random Slope	$\eta_{ij} = \beta_{0j} + \beta_1 X_{1ij} + \ldots + \beta_{15} X_{15ij}$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $\beta_{1j} = \gamma_{10} + u_{1j}$, $\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma_{u01} \\ \sigma_{u01} & \sigma^2_{u1} \end{pmatrix} \right)$.
F	Random Intercept and Random Slope with Cross-Level Interaction	$\eta_{ij} = \beta_{0j} + \beta_{1j} X_{1ij} + \beta_2 X_{2ij} + \ldots + \beta_{15} X_{15ij}$, $\beta_{0j} = \gamma_{00} + \gamma_{01} Z_{1j} + \ldots + \gamma_{05} Z_{5j} + u_{0j}$, $\beta_{1j} = \gamma_{10} + \gamma_{11} Z_{1j} + u_{1j}$, $\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma_{u01} \\ \sigma_{u01} & \sigma^2_{u1} \end{pmatrix} \right)$.

The data-generating process incorporated both level-1 and level-2 covariates to reflect the hierarchical nature of the models. At level-1, X_{pij} represents a matrix of either 15 covariates (Models A, E, and F) or 17 convariates (Models B, C, and D), including both continuous and dichotomous variables. Continuous covariates were sampled from a multivariate normal distribution with mean zero and a variance-covariance matrix Σ, while dichotomous covariates followed a binomial distribution, $B(0.5)$. To simulate realistic multicollinearity, four continuous covariates were generated jointly using a multivariate normal distribution. The fixed effects for level-1 predictors (β_1 through β_{17}) were sampled from $\mathcal{N}(0, 1)$, resulting in a range of effect sizes from negligible to significant (Table 3.3).

At level-2, Z_{qij} represents a matrix of 5 continuous predictors of the level-2 intercept, sampled from $\mathcal{N}(0, 1)$. The overall intercept γ_{00} was set to zero, while $\gamma_{01}, \gamma_{02}, \gamma_{03}, \gamma_{04}, \gamma_{05}$ were

sampled from $\mathcal{N}(0, 1)$. Additionally, γ_{11} was set to -0.1 in Model F to model the strength of the cross-level interaction, where the level-2 predictor Z_{1j} influenced the random slope β_{1j}.

For Models A-D, the random intercept u_{0j}, which captures deviations from the fixed population intercept, was modeled as $u_{0j} \sim \mathcal{N}(0, \sigma^2_{u0})$. For Models E and F, the relationship between level-1 predictors and the outcome (β_{1j}) varied across Level-2 units. This was represented as:

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma_{u01} \\ \sigma_{u01} & \sigma^2_{u1} \end{pmatrix}\right) \tag{3.1}$$

where σ_{u01} reflects the covariance between the random intercept (σ^2_{u0}) and the random slope (σ^2_{u1}). These variance components (σ^2_{u0}, σ^2_{u1}, and σ_{u01}) were varied systematically across simulation conditions to evaluate their effects on model performance.

Table 3.3: The true parameter values for all models, including fixed effects and variance components used in data generation.

Level-1 Coefficients	**Values**
β_1	-0.7
β_2	1
β_3	0.8
β_4	-1.1
β_5	0.3
β_6	-0.3
β_7	-1.4
β_8	-0.4
β_9	-0.2
β_{10}	0.6
β_{11}	1.2
β_{12}	-1.2
β_{13}	0.7
β_{14}	0.4
β_{15}	0.7
β_{16}	-0.2
β_{17}	0.1
Level-2 Coefficients	**Values**
γ_{00}	0
γ_{01}	-1.3
γ_{02}	-0.1
γ_{03}	0.7
γ_{04}	0.5
γ_{05}	0
γ_{11}	-0.1
Level-2 Variance Components	**Values**
σ^2_{u1}	0.25
σ_{u01}	0.001, 0.14, 0.2
σ^2_{u0}	0.0001, 2, 4

The second manipulated factor considered in the simulation study controls the amount of residual level-2 variance within the generated data. This factor reflects the variability in outcomes that is not explained by the predictors which is believed to have a significant impact on the performance of the standard and multilevel algorithms. Additionally, this factor contributes to

understanding how well the algorithms perform under different degrees of noise and aids in determining their robustness to variations in data complexity and the influence of the nested structure. Three distinct levels were selected for the residual Level-2 variance condition, spanning from nearly zero to a relatively high value for educational research studies. To control the level of residual level-2 variance, σ^2_{uo}, σ^2_{u1}, and σ_{u01} were manipulated to introduce varying amounts of between-cluster variance, creating scenarios that reflect typical research contexts. The first scenario, $\sigma^2_{uo} = 0.001$, emulates cases where researchers have access to level-2 variables capable of explaining nearly all the between-cluster variance, resulting in a nearly zero residual ICC. This scenario aligns with findings from prior research indicating that the inclusion of covariates can account for a substantial portion of between-school variance, often reducing the ICC significantly (Hedges and Hedberg, 2007). The second scenario, $\sigma^2_{uo} = 2$, emulates situations where researchers have a set of predictors that can account for some of the between-cluster variance, but there is still some which remains unexplained. Finally, the third scenario, $\sigma^2_{uo} = 4$, represents conditions where researchers lack strong predictors for the between-cluster variance, leading to a higher residual level-2 variance. These scenarios provide a spectrum of ICC values that encompass typical conditions encountered in education research (Hedges and Hedberg, 2007). Figure 3.1 shows a boxplot illustrating how increasing σ^2_{u0} affects the distribution of ICCs in data generated from a multilevel logistic regression without level-1 and level-2 predictors. In contrast, including level-1 and level-2 predictors in the data generation process moderates the effects of residual level-2 variance. A pilot study determined the values of σ^2_{u0} that achieved the desired levels of between-cluster variability for data-generating models A-F: when $\sigma^2_{u0} = 0.001$, the average empirical ICC was 0.20; for $\sigma^2_{u0} = 2$, it increased to 0.29; and for $\sigma^2_{u0} = 4$, it reached 0.37.

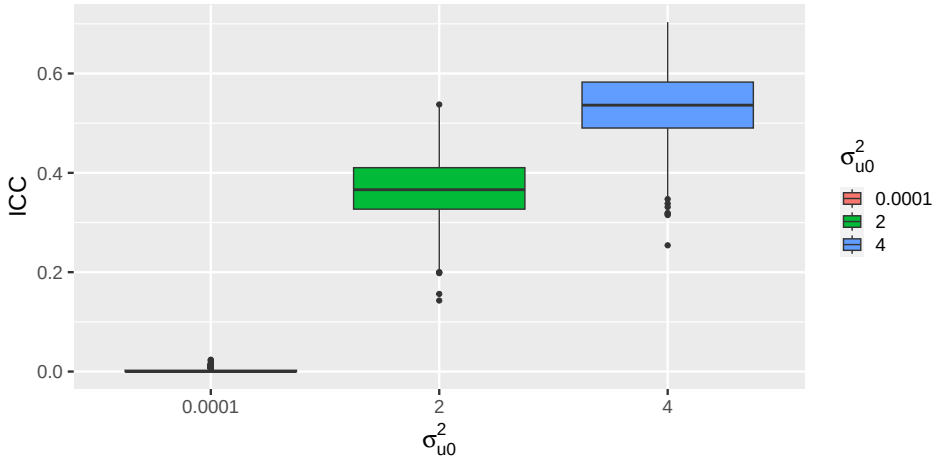

Figure 3.1: Distribution of Intraclass Correlation Coefficients (ICCs) by levels of residual level-2 variance (σ^2_{u0}) in data generated from a multilevel logistic regression without level-1 and level-2 predictors. Each box represents the variability in ICC estimates across 500 datasets for each condition of σ^2_{u0} (0.001, 2, 4). Higher values of σ^2_{u0} correspond to greater between-cluster variability, resulting in increased ICCs.

The third manipulated factor is the sample size. Here, the study considers variations in both the number of clusters and cluster sizes. Prior research has predominantly focused on the impact of sample size on parameter estimates for statistical inference, consequently, there is a notable scarcity of guidelines about optimal sample sizes for classification purposes. Mangino (2021) and Mangino et al. (2022) conducted simulation studies that systematically addressed sample size as a manipulated factor in classification settings. Mangino and Finch's 2021 study covered three cluster size levels: 100, 500, 1000, and five levels of the numbers of clusters: 10, 20, 30, 50, 100. Their 2022 study, in contrast, focused on smaller sample sizes, featuring three cluster size levels: 10, 50, 100, and three levels of the numbers of clusters: 10, 50, 100. The 2021 study found that an increase in both the number of clusters and number of cases within each cluster led to an improvement in all accuracy measures. However, the 2022 study revealed that their results

largely were not affected by sample size, except that an increase in cluster size slightly decreased the observed small cluster recovery rate while slightly increasing AUC and binary cross entropy. While not specific to classification, McNeish and Stapleton (2016) demonstrated that multilevel models, although potentially underpowered, can provide reliable parameter estimates with as few as 10 clusters. Guidelines have also been provided for the number of level-2 clusters necessary for multilevel models to fit the data appropriately ranging from 30 to 50, 40 to 60 and, 60 to 100 (Maas and Hox, 2005; McNeish and Kelley, 2019; Meuleman and Billiet, 2009).

Given the existing literature and previous simulation studies, the present study considered three levels of the number of clusters: 30, 50 and 100 and two levels of cluster sizes: 50, 100, in line with Kreft and De Leeuw (1998), Maas and Hox (2005) and McNeish and Stapleton (2016) methodologies. It should be noted that initially, a condition which considered 500 individuals per cluster was considered; however, this proved to be too computationally intensive and was therefore dropped from the current study.

The final manipulated factor in this study is how the training and testing datasets are split. When working with multilevel models, predicting new outcomes involves two distinct scenarios. The first scenario involves predicting the outcome for a new subject, denoted as i^*, where no prior information about the random effects is available. This situation arises when the subject i^* belongs to a new cluster j^*, that was not part of the training dataset. For instance, this could involve predicting whether a new student in an entirely new classroom- one that was not included in the training data- will graduate. In this case, predictions rely solely on the fixed effects component of the model because there is no information about the cluster-specific random effects. This type of training-testing split, known as a cluster-level split, tests the model's ability to generalize to new, unseen clusters and provides insights into how well a model may perform in real-world

settings where new clusters are likely to appear, such as new classrooms or schools being added to an educational system.

In contrast, the second scenario focuses on predicting the outcome for a new subject i^* within a cluster j that was part of the training process. For example, this could involve predicting if a new student in a classroom already represented in the training dataset will graduate. In this scenario, the estimated random effects for cluster j are available, allowing the prediction to include both the fixed effects and the cluster-specific random effect of cluster j. This type of split, known as a within-cluster split, simulates a situation where predictions are needed for new members of existing clusters, such as incoming students in a classroom for which historical data is available.

The distinction between these scenarios is important when evaluating prediction performance. The simulation study considers both scenarios by 1) splitting the data within cluster and 2) splitting the data at the cluster-level. Evaluating models under these different conditions helps assess their generalizability and effectiveness for real-world applications where predictions may be needed across or within clusters.

Finally, an additional condition specific to the within-cluster training testing split conditions was considered to further assess model performance. This condition evaluated whether treating cluster IDs as fixed effects in a standard fixed effects model could sufficiently capture the underlying hierarchical structure without the need for a multilevel model. For example, this could involve including classroom IDs as dummy variables to determine whether they adequately account for variability between clusters. Since the between-cluster training-testing split inherently involves distinct cluster IDs in the training and testing sets, this condition is only applicable to the within-cluster split. By exploring this condition, the study examines whether simpler fixed effects

models can effectively approximate the hierarchical nature of the data or if multilevel models are necessary for capturing complex structures.

3.1.2 Classification Algorithms

A range of prediction algorithms were compared in this study, including both standard (single-level) and multilevel algorithms. Specifically, the following algorithms are evaluated: logistic regression, lasso, decision trees, XGBoost, random forests, support vector machines with a linear kernel, support vector machines with a polynomial kernel, artificial neural networks, generalized linear mixed effects model, multilevel lasso, multilevel decision trees, and multilevel random forests, Table 3.4. These twelve algorithms were applied to data generated across each simulation condition outlined in Section 3.1.2, using an 80:20 training-to-testing split. Most methods required the selection of hyperparameters, except for logistic regression and GLMM. Cross-validation was employed as the primary method for hyperparameter tuning due to its reliability in assessing model generalizability and minimizing overfitting (Kohavi, 1995). The standard algorithms were chosen because they represent commonly used methods in predictive modeling (Luan and Tsai, 2021). In contrast, the multilevel algorithms were selected to represent the prediction algorithms that have been extended to incorporate mixed effects. Additionally, only multilevel algorithms that successfully converged during a pilot study were included. This set of algorithms expands upon previous studies that have compared standard and multilevel classification algorithms (Cannistrà et al., 2020; Fontana et al., 2021; Hajjem et al., 2017; Mangino et al., 2022; Ngufor et al., 2019; Pellagatti et al., 2021).

Table 3.4: Algorithms considered in the study

Standard Algorithm	Multilevel Extensions
Logistic regression	Generalized Linear Mixed Model (GLMM)
Lasso	Multilevel Lasso (GLMMLasso)
Decision trees	Multilevel Classification and Regression Trees (M-CART)
Random forests	The Binary Mixed Model (BiMM) Forest
XGBoost	
Support Vector Machines - Linear Kernel	
Support Vector Machines - Polynomial Kernel	
Artificial Neural Networks	

3.1.3 Evaluation Metrics

To evaluate the performance of each algorithm across each manipulated condition of the simulation study and the subsequent empirical analysis, several outcome metrics were reported including: the overall classification accuracy, sensitivity, specificity, AUC-ROC, and F1 score. Classification accuracy reflects the percentage of observations correctly classified by the prediction algorithm and is calculated as the sum of true positives and true negatives divided by the total number of cases: $\frac{\text{TN+ TP}}{\text{TP+ TN+FP+FN}}$. The sensitivity, also known as recall or the "true positive rate", quantifies an algorithm's ability to correctly identify positive outcomes from the actual positive class. It represents the ratio of true positives to the sum of true positives and false negatives: $\frac{\text{TP}}{\text{TP+FN}}$. The specificity of an algorithm can be thought of as the "true negative rate" - the percentage of negative outcomes the model detects: $\frac{\text{TN}}{\text{TN+FP}}$. Sensitivity and specificity offer valuable insights when working with an imbalanced dataset, where one class significantly outweighs the other. AUC-ROC measures the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. A higher AUC-ROC value indicates a better-performing model with improved discrimination power. The F1 score, also referred to as the F-score or the F-measure, considers the trade-off between precision p (the number of true positives divided by the

number of true positives and false negatives) and its recall r (the number of true positives divided by the number of true positives and false positives). The F1 score represents the harmonic mean of precision and recall: $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$, providing a single metric to assess a classifier's performance. Notably, the F1 score is particularly useful when dealing with imbalanced datasets. These metrics collectively provide a comprehensive evaluation of each algorithm's performance, offering different perspectives of what it may mean to be effective in each unique research setting. They highlight each algorithm's strengths and weaknesses, ultimately aiding in the selection of the most suitable models for specific prediction tasks encountered in educational research.

3.2 Empirical Illustration

In addition to the simulation study, an illustrative example was conducted to compare the performance of the algorithms and to demonstrate their practical implementation. The Maryland Longitudinal Data System (MLDS) served as the data source for these analyses. MLDS is a statewide data repository that links data across the Maryland State Department of Education, the Maryland Higher Education Commission and the Department of Labor, Licensing and Registration. These data provide individual-level records based on demographic and academic information, encompassing data on attendance, assessments, course enrollment, retention, and degree attainment.

This study utilized the MLDS data to examine how well standard and multilevel algorithms predicted STEM major persistence. The analysis incorporated a variety of predictors at both the student and post-secondary institution level to account for individual and contextual influences on STEM persistence. The outcome variable, STEM persistence, was defined as a binary indicator

of whether a student earned a STEM degree within six years of initial enrollment (full sample) and within four years of initial enrollment (imbalanced dataset). STEM programs were identified using the U.S. Department of Homeland Security's STEM Designated Degree Program List. Predictors included program-specific details, such as the STEM major pursued at enrollment, as well as incoming academic attributes like standardized test scores. Demographic features, such as race, sex, and Pell Grant status, were included to capture individual-level variation, while institutional factors provided insight into the broader postsecondary environment (e.g., average GPA at the post-secondary institution level and the size of the post-secondary institution) (Table 3.5). All continuous variables were scaled to ensure that each predictor contributed equally to the model and to prevent variables with larger ranges from disproportionately influencing the predictions. The empirical analysis also investigated the ability of the algorithms to handle class imbalance, a common challenge in predictive modeling. While the full dataset had a relatively balanced class distribution (43% dropouts vs. 57% non-dropouts), the secondary analysis focused on a highly imbalanced subset. This subset included Black students who received Pell Grants, with only 15% obtaining a STEM degree in four years. Such imbalance often results in model bias toward the majority class, inflating overall accuracy at the expense of correctly classifying the minority group (Johnson and Khoshgoftaar, 2019). This analysis aimed to evaluate the algorithms' robustness in both balanced and imbalanced scenarios.

Table 3.5: Defined variables used in the empirical analysis

Category	Features	Description
Outcome Variable	STEM Persistence	Dichotomous indicator variable of obtaining a STEM degree within 6 years of the first year of enrollment (full dataset) or within 4 years of the first year of enrollment (imbalanced dataset). A STEM program is defined by The U.S. Department of Homeland Security (DHS) STEM Designated Degree Program List - a complete list of fields of study that DHS considers to be science, technology, engineering, or mathematics (STEM) fields of study.
Program Information	STEM Program	The STEM major a student pursued at the time of postsecondary enrollment
	Freshman Year GPA	Student's GPA after the first year of postsecondary enrollment
	Freshman Year Credits Earned	The total number of credits a student earned during their first year of postsecondary enrollment

Incoming Attributes	SAT or ACT Score	Student's maximum combined SAT or ACT score reported
Demographics	Sex	Binary sex status indicator for Male/Female
	Race	A 4-level variable indicating a student's race (White, Black, Asian, Other)
	Hispanic	Binary indicator that a student is Hispanic
	Filed a FAFSA	Binary indicator variable if a student filed a FAFSA
	Pell Grant	Binary indicator variable if a student received a Pell Grant
Institution Characteristics	College GPA	The average GPA for the fall semester at the college level
	College credits earned	The average credits earned for the fall semester at the college level
	College size	The number of students enrolled in each college during the fall semester
	Public	Binary indicator variable if the college is public vs a private institution

3.3 Summary

This research was structured in a two-fold approach: a simulation study and two empirical analyses of datasets from the Maryland Longitudinal Data System. The simulation study manipulated factors such as the data generating model, the amount of residual level-2 variance, sample size (number of clusters and individuals per cluster), and the training: testing split. Twelve prediction algorithms, including both standard and multilevel algorithms, were evaluated based on multiple outcome metrics. The empirical analyses used real-world data from MLDS to predict which students were at risk of dropping out of a STEM major vs those that obtained their STEM degree and investigated algorithm performance in the context of sparse outcomes. The empirical analyses serve as a practical illustration of the simulation study's findings, demonstrating the application of these prediction algorithms in the context of educational research. Finding from these studies provide insights for researchers and practitioners in the field of education by comparing the performance of standard and multilevel classification algorithms in predicting outcomes from hierarchical datasets.

Chapter 4: Simulation Results

4.1 Simulation Study

The results from this simulation study provide a comprehensive view of how different machine learning models, both standard and multilevel, perform under varying conditions that reflect the complexities of real-world hierarchical data. The chapter begins with a discussion of overall model convergence, followed by the effect of the training-testing split (within-cluster vs. between-cluster) on each model, and finally the effects of the manipulated simulation conditions, including residual level-2 variance (σ^2_{u0}), number of clusters (J), individuals per cluster (n_j), and data-generating processes (DGP). These results are explored through graphical illustrations of performance trends to provide an intuitive understanding of the models' behavior under different conditions.

To complement these visualizations, additional ANOVA results are summarized in Table A.1. This table provides a breakdown of the main effects and interactions of the manipulated factors, highlighting their statistical and practical significance across both the within-cluster and between-cluster split conditions.

Given the balanced nature of the simulated datasets, accuracy will primarily be used to visualize model performance, given its similarity to AUC and F1 scores across conditions, though sensitivity and specificity are also considered where relevant. AUC and F1 score plots are avail-

able in the appendix.

4.1.1 Convergence

Convergence is an important aspect of model performance, especially in simulation studies where certain models may struggle to converge under specific conditions. In this study, the convergence rates of each algorithm were calculated as the proportion of times an algorithm successfully converged within a set computational time limit or a specified number of iterations across 500 replications for each simulation cell. A variety of optimizers were used to facilitate convergence under these constraints.

Across all models, convergence rates were generally high, with most algorithms achieving near perfect convergence ($\geq 99\%$) in the majority of conditions, as shown in Figure 4.1 which displays convergence rates across all conditions for each model. However, variability in convergence was observed in three models: both SVM models and GLMM. Residual level-2 variance, σ^2_{u0}, had the strongest impact on GLMM's convergence. GLMM achieved high convergence in conditions with low to moderate residual level-2 variance ($\sigma^2_{u0} = 0.0001$, 95% average convergence; $\sigma^2_{u0} = 2$, 96% average convergence across all other conditions). However, convergence rates declined slightly under high residual variance ($\sigma^2_{u0} = 4$, 94% average convergence across all conditions). The decrease was particularly pronounced in certain simulation cells with $\sigma^2_{u0} = 4$, such as when there were 30 clusters with 50 individuals per cluster, a between-cluster split, and data generated using Model F, where convergence fell to 82%. Although the average decrease was minimal, it suggests that the computational demands of estimating complex cluster-level effects under high residual variance can challenge the stability of GLMM.

In contrast, the SVM models encountered convergence issues primarily related to the time-to-convergence limit when sample sizes were large. SVMs require significant computational resources for large datasets because they calculate the relationship between every pair of data points to find the optimal decision boundary. This process involves solving a complex optimization problem, which becomes increasingly time-consuming as the number of samples grows. To ensure the feasibility of the study within the University of Maryland's High-Performance Computing (HPC) resource allocation, a two-hour time limit was imposed for model fitting. This limit balanced the practical constraints of computational resources with the need to evaluate a wide range of models across all simulation conditions. The two-hour window was determined based on pilot studies and was sufficient for most scenarios. Models failing to converge within this window were classified as "nonconvergence". Despite these challenges, SVMs converged within two hours in most conditions, with an overall average convergence rate of 99%. However, for conditions with 100 clusters and 100 individuals per cluster, SVM with a polynomial kernel converged 98% of the time, while the SVM with a linear kernel converged only 94% of the time. While convergence variability was minimal overall, these findings highlight the importance of understanding computational constraints and model-specific requirements in real-world applications. Researchers should be aware that models like GLMM and SVM may require more advanced optimization techniques or additional computational resources, particularly in complex data scenarios involving large samples or high residual variance.

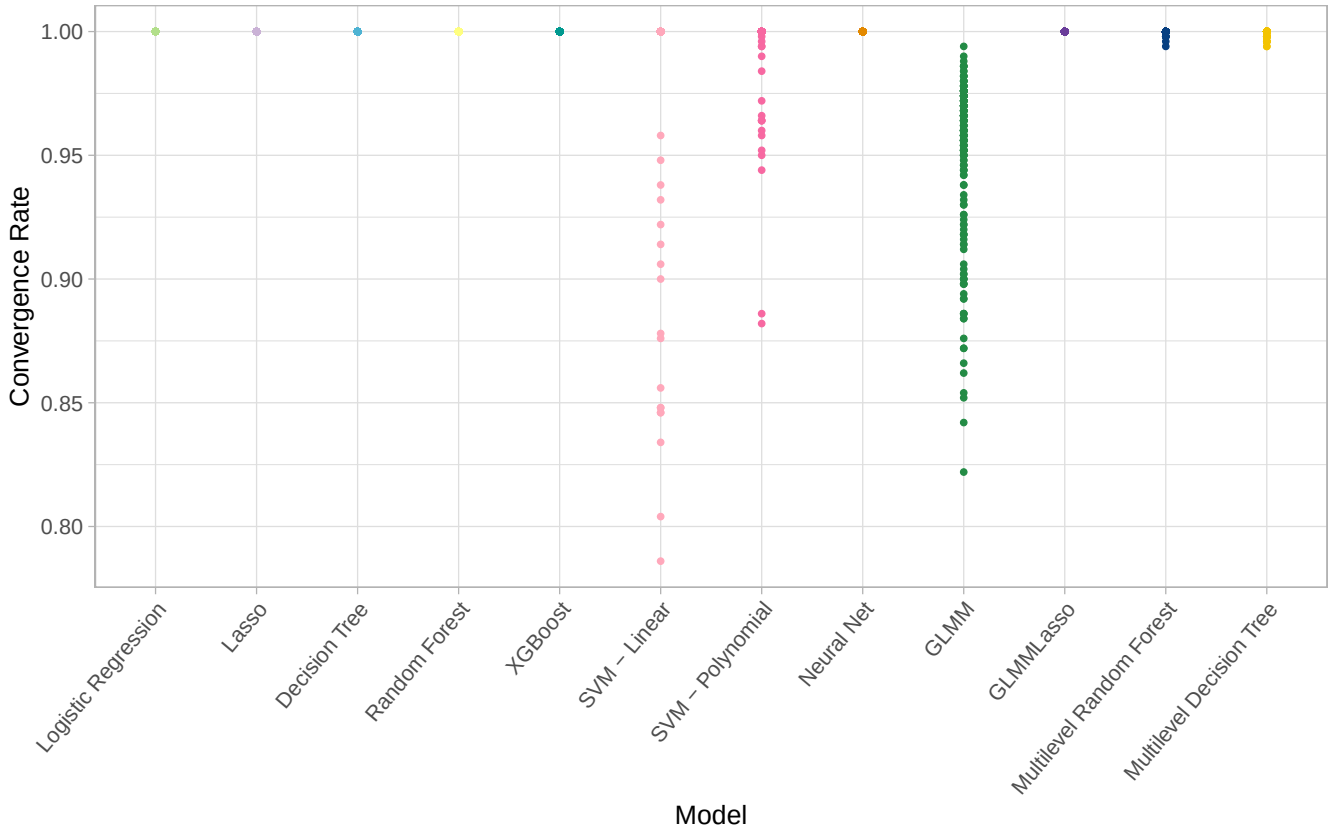

Figure 4.1: Convergence rates for each model across 500 replications. Each dot represents a simulation cell.

4.1.2 Effect of Training-Testing Split

Model performance was evaluated under two types of training-testing splits: the within-cluster split, where individuals from each cluster were represented in both training and testing datasets, and the between-cluster split, where distinct clusters were assigned exclusively to either the training or testing dataset. The distinction between these training-testing split conditions offers insights into each model's ability to predict new observations for known clusters (within-cluster split) as well as their ability to generalize across entirely new clusters (between-cluster split). Performance metrics for models under within-cluster and between-cluster training-testing splits were averaged across all other manipulated factors, including residual level-2 variance (σ^2_{u0}), the number of clusters (J), the number of individuals per cluster (n_j), and the data-generating processes (DGP). This aggregation allowed for the evaluation of overall trends in model generalizability across the different training testing split conditions

As shown in Figures 4.2, 4.3, and 4.4, most models exhibited higher performance metrics (e.g., accuracy, sensitivity, and specificity) in the within-cluster split compared to the between-cluster split. This result aligns with theoretical expectations, as the within-cluster split provides training data that more closely resembles the testing data allowing models to learn and leverage intra-cluster information when making predictions.

Logistic Regression, Lasso, and SVM with a linear kernel demonstrated relatively consistent performance across both split conditions, suggesting that these models may be useful in scenarios where predictions need to be made for both new observations within known clusters and for entirely new clusters. While these models did not achieve the highest performance in the within-cluster split condition compared to multilevel models like GLMM and GLMMLasso,

their stability across split conditions is notable. For example, in the within-cluster split, Logistic Regression achieved an average accuracy of 0.82, sensitivity of 0.84, and specificity of 0.79. In the more challenging between-cluster split, these metrics declined only slightly to an average accuracy of 0.81, sensitivity of 0.83, and specificity of 0.77. Lasso and SVM with a linear kernel exhibited similarly stable patterns: Lasso achieved an average accuracy of 0.82 in the within-cluster split and 0.81 in the between-cluster split, while SVM with a linear kernel also achieved 0.82 average accuracy in the within-cluster split and 0.81 in the between-cluster split. Similarly small declines were observed in sensitivity and specificity for both models (Figures 4.2, 4.3, and 4.4).

In contrast, models such as Decision Tree, Random Forest, and SVM with a polynomial kernel showed greater sensitivity to the between-cluster split condition. Random Forest, for instance, experienced a notable decline in performance: average accuracy dropped from 0.81 in the within-cluster split to 0.76 in the between-cluster split, and specificity declined from 0.74 to 0.67. Similarly, Decision Tree showed a decrease in average accuracy from 0.66 to 0.62 and specificity from 0.54 to 0.50. XGBoost and Neural Nets exhibited increased variability in their prediction performances, as indicated by the larger ranges observed in their accuracy, sensitivity, and specificity boxplots. For example, in the between-cluster split, the range of accuracy for XGBoost spanned from 0.72 to 0.85, compared to 0.81 to 0.85 in the within-cluster split. Similarly, Neural Nets showed an accuracy range of 0.70 to 0.85 in the between-cluster split, whereas in the within-cluster split, the range was narrower, from 0.79 to 0.85. This increased variability suggests that while these models can perform well in some scenarios, their predictions may be less reliable across different data splits. Consequently, while models like XGBoost and Neural Nets perform well in within-cluster splits, simpler models such as Logistic Regression and Lasso

may offer more consistent performance when dealing with between-cluster data splits.

In the within-cluster split condition, GLMM and GLMMLasso consistently outperformed other models across all metrics on average. GLMMLasso achieved the highest average classification accuracy (0.85) and sensitivity (0.87), while GLMM had the highest average specificity (0.88), F1-score (0.84), and AUC (0.85). Additionally, these models also demonstrated the highest stability, as evidenced by the narrower ranges observed in their prediction metrics (Figures 4.2, 4.3, and 4.4). For instance, the accuracy range for both GLMM and GLMMLasso in the within-cluster split was only 0.83 to 0.86, compared to wider ranges seen in the other models. This stability and higher average performance metrics suggest that GLMM and GLMMLasso effectively capture cluster-level dependencies, allowing them to make consistent and improved predictions by accounting for both individual and cluster-level variance components. Even in the more challenging between-cluster split condition, GLMMLasso and GLMM remained among the top performers, each achieving an average accuracy of 0.80, as shown in Figure 4.2.

These findings suggest that the choice of model should depend on the nature of the data, the type of prediction required, and the intended application. For scenarios where clusters remain consistent across training and testing (e.g., predicting a new student's performance in a classroom which was included in model training), multilevel models like GLMM and GLMMLasso are particularly valuable. Their ability to capture both individual- and cluster-level dependencies results in higher average performance and more stable predictions. However, for applications involving new clusters (e.g., predicting outcomes for students in new schools), simpler models such as Logistic Regression, Lasso, and SVM with a linear kernel may offer more consistent and robust performance. These models exhibited relatively small declines in performance metrics between the within-cluster and between-cluster splits.

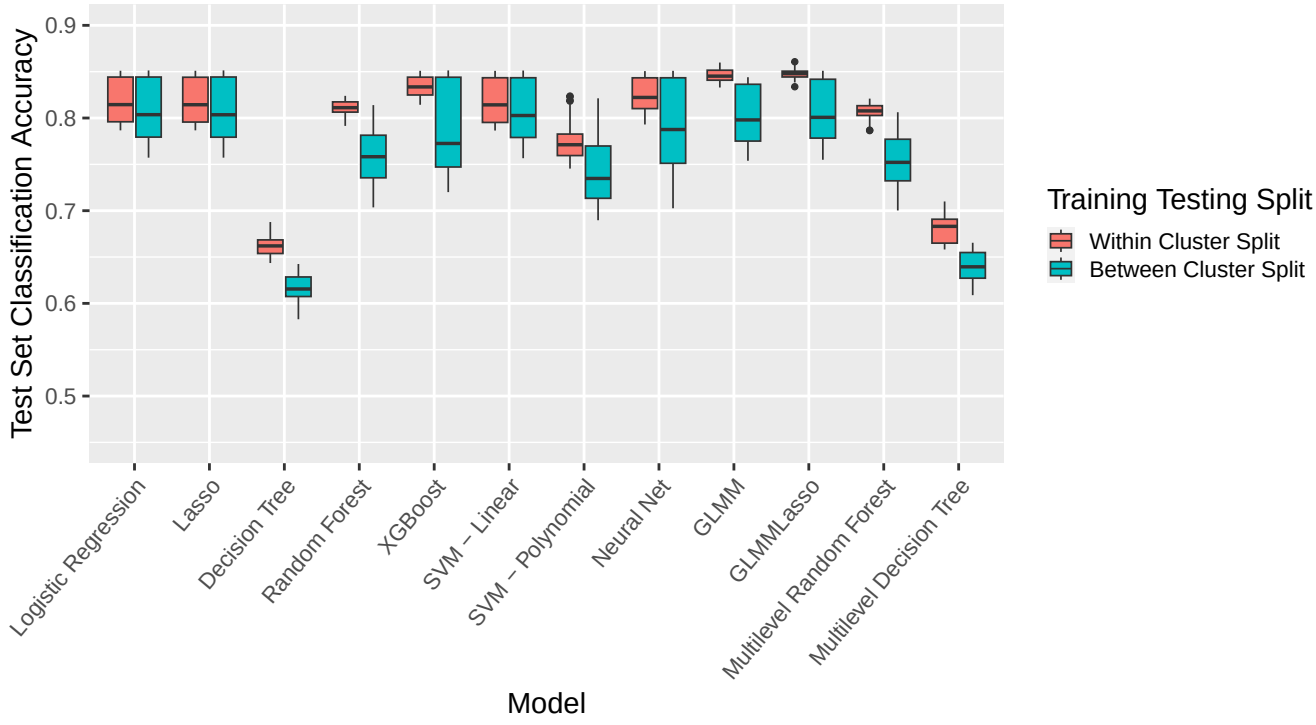

Figure 4.2: Classification accuracy across models for the within-cluster split and the between-cluster split condition. The bars are grouped by split condition, aggregating performance across all other manipulated simulation factors, including residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

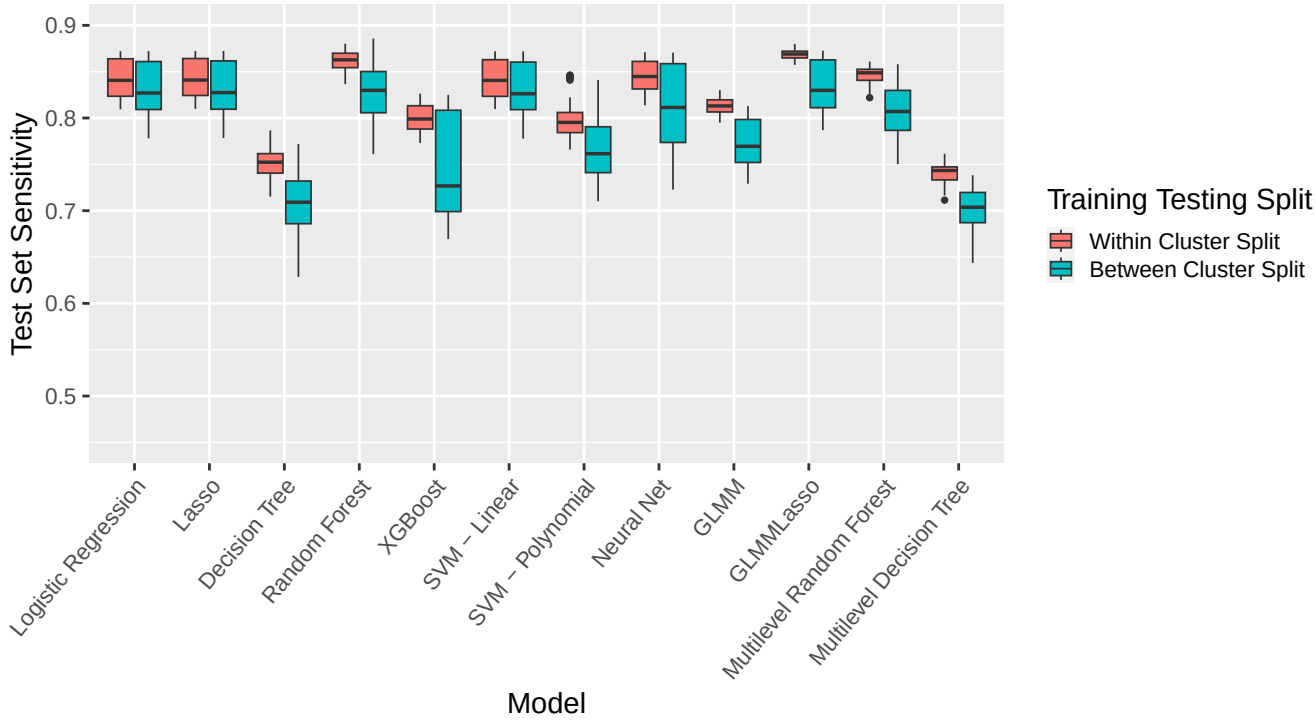

Figure 4.3: Average sensitivity (true positive rate) for each model under the within-cluster and between-cluster split condition. Bars for each model are grouped by split condition, with sensitivity values aggregated across all other manipulated simulation factors, such as residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

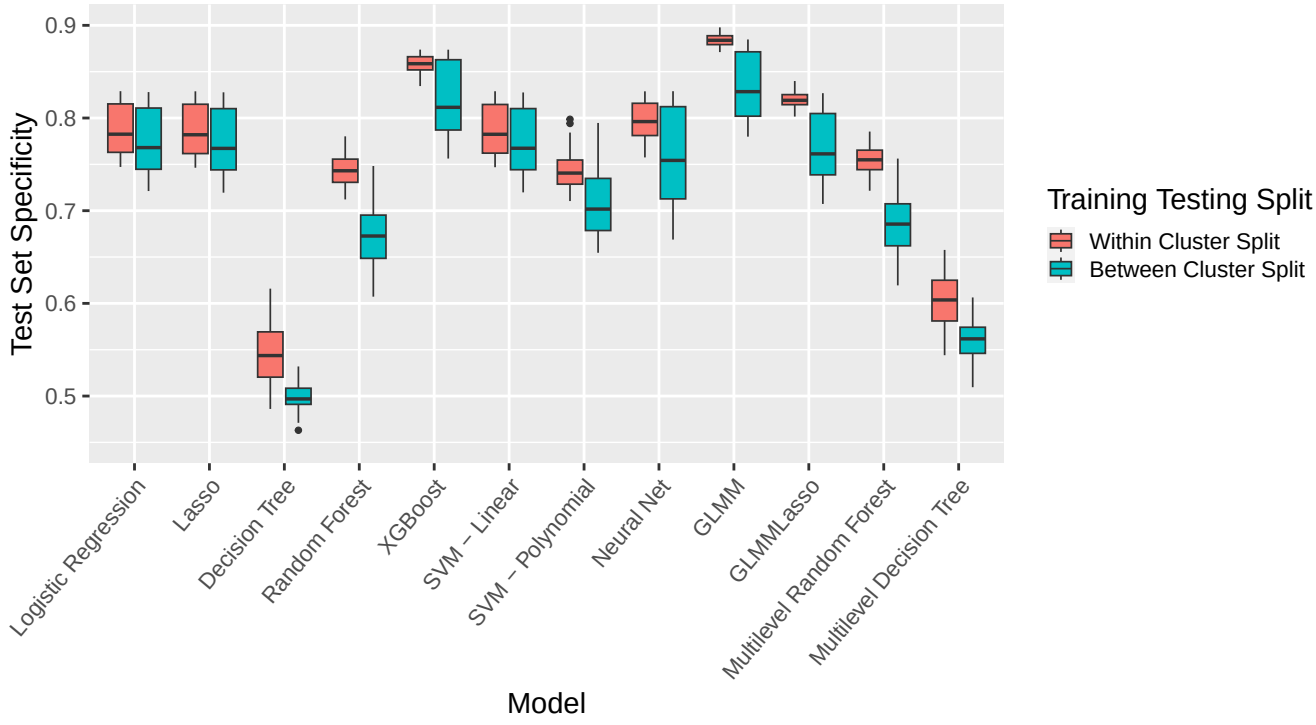

Figure 4.4: Average specificity (true negative rate) achieved by each model under the within-cluster and between-cluster splits conditions. The grouped bars aggregate specificity values across all other manipulated simulation factors, including residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

4.1.3 Effects of Individual Factors within the Training-Testing Split Context

4.1.3.1 Effect of Residual Level-2 Variance: σ^2_{u0}

Residual level-2 variance, σ^2_{u0}, controls the additional variability between clusters beyond the effects of level-2 predictors in the data-generating model. Three levels were evaluated in this study: $\sigma^2_{u0} = 0.0001$, $\sigma^2_{u0} = 2$, and $\sigma^2_{u0} = 4$. Accuracy results are displayed in Figure 4.5, while sensitivity, specificity, AUC, and F1 score results are provided in the Appendix (Figures A.3, A.4, A.5, and A.6). These results are aggregated across all other manipulated simulation factors (number of clusters, number of individuals per cluster, and the data-generating process) to provide a global summary of how models handle increasing residual cluster-level variability.

In the within-cluster split, models generally maintained stable accuracy as σ^2_{u0} increased. Standard models, such as Logistic Regression, Lasso, both SVM models, and Neural Net, showed slight declines in accuracy as σ^2_{u0} increased. For example, all three of these models' average classification accuracy decreased from 0.85 when $\sigma^2_{u0} = 0.0001$ to 0.79 when $\sigma^2_{u0} = 4$, reflecting a 6% reduction. These patterns suggest that standard models may struggle to handle the additional cluster-level variability introduced by higher residual level-2 variance.

In contrast, multilevel models (GLMM, GLMMLasso, Multilevel Random Forest, and Multilevel Decision Tree) generally improved or maintained their performance as σ^2_{u0} increased. For instance, GLMM's accuracy increased from 0.84 at $\sigma^2_{u0} = 0.0001$ to 0.85 at $\sigma^2_{u0} = 4$. Similarly, Multilevel Random Forest's accuracy rose from 0.80 to 0.81. These trends highlight their ability to model and leverage cluster-level variance effectively. Interestingly, ensemble methods such as Random Forest and XGBoost also showed relatively stable performance across σ^2_{u0} lev-

els. Random Forest showed a slight increase in accuracy from 0.81 when $\sigma^2_{u0} = 0.0001$ to 0.82 when $\sigma^2_{u0} = 4$, while XGBoost only saw a 2% decrease in average accuracy (0.85 to 0.83) as σ^2_{u0} increased. This may suggest that these ensemble methods may be indirectly learning cluster specific information, allowing them to perform well in within-cluster splits without explicitly modeling cluster-level random effects.

The effect of increasing σ^2_{u0} was more pronounced in the between-cluster split, where models lacked access to within-cluster information during testing. Accuracy consistently declined for all models as σ^2_{u0} increased. Logistic Regression's accuracy decreased from 0.85 at $\sigma^2_{u0} = 0.0001$ to 0.77 at $\sigma^2_{u0} = 4$, a 8% reduction. Lasso and SVM with a linear kernel followed similar patterns, both also dropping from approximately 0.85 to 0.77.

Multilevel models also experienced notable declines in accuracy. For instance, both GLMM's and GLMMLasso's accuracy decreased from 0.84 at $\sigma^2_{u0} = 0.0001$ to 0.77 at $\sigma^2_{u0} = 4$. These declines align with theoretical expectations, as higher σ^2_{u0} amplifies the clustering effect, making it more challenging to generalize across new clusters.

Ensemble methods such as Random Forest and XGBoost exhibited very steep declines in accuracy. For Random Forest, accuracy decreased from 0.79 to 0.73, and for XGBoost, from 0.85 to 0.74. This suggests that these methods may be less effective in generalizing to entirely new clusters when residual level-2 variance is high.

Overall, these results highlight the potential advantage of multilevel models under conditions of high between-cluster variance when training and testing data are split within clusters. However, as data become more and more cluster-specific and clusters are separated across training and testing splits, accuracy declines across all models, regardless of their ability to model cluster-level effects.

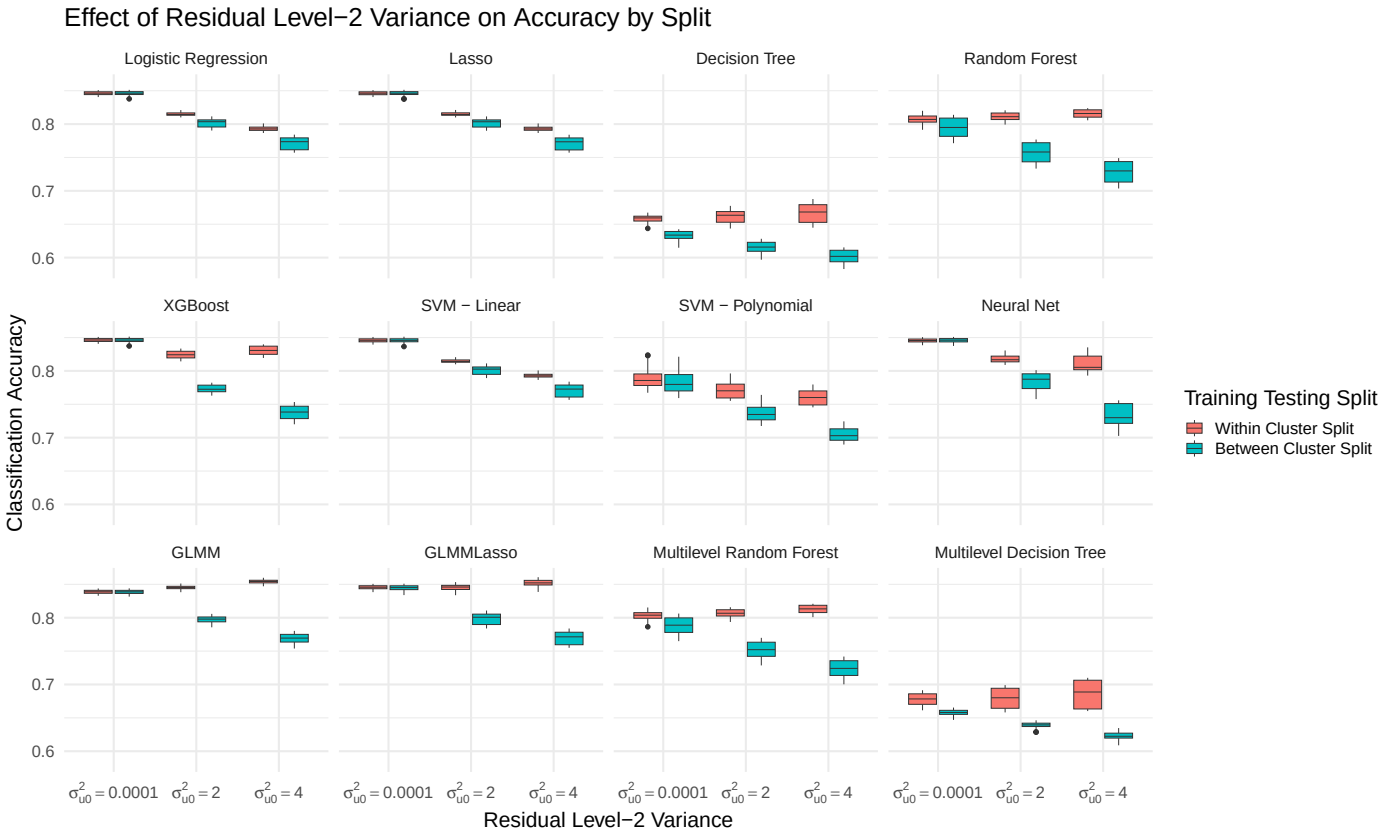

Figure 4.5: Classification accuracy by model, training testing split, and across levels of residual level-2 variance (σ^2_{u0}). Results are aggregated across all other manipulated simulation factors, including the number of clusters, individuals per cluster, and the data-generating process.

4.1.3.2 Effect of Number of Clusters: J

The number of clusters J was varied across three levels: 30, 50, and 100. This factor exhibited notable trends in model performance, particularly in the between-cluster split, while the within-cluster split showed smaller effects. Results for classification accuracy and sensitivity are presented in Figures 4.6 and 4.7. The results presented in this section are aggregated over residual level-2 variance (σ^2_{u0}), the number of individuals per cluster (n_j), and the data-generating processes (DGP).

In the within-cluster split, classification accuracy showed minimal change as the number of clusters increased for most models. Logistic Regression, Lasso, and Neural Net maintained stable accuracy of 0.82 for all levels of the number of clusters. However, standard decision trees and multilevel decision trees showed a decrease in accuracy as the number of clusters J increased. For example, standard decision trees saw a drop in mean accuracy from 0.673 for $J = 30$ to 0.652 for $J = 100$, while multilevel decision trees declined from 0.697 to 0.664. This pattern indicates potential overfitting to cluster-specific structures in the training data, which decision tree-based methods are often criticized for.

In contrast, the between-cluster split showed a more consistent improvement in accuracy as the number of clusters increased. This trend was strongest for the standard and multilevel random forest models. Standard random forests, for instance, increased their mean accuracy from 0.74 for $J = 30$ to 0.78 for $J = 100$, while multilevel random forests improved from 0.74 to 0.77. This may suggest that having a larger number of clusters improves generalizability, even when clusters in the testing set are distinct from those in the training set. With more clusters available for training, models are exposed to a broader range of patterns, potentially enabling more accurate

predictions on unseen clusters.

Sensitivity results largely mirrored the trends observed in accuracy (Figure 4.7). Standard decision trees (and, to a lesser extent, multilevel decision trees) did not exhibit a decline in sensitivity with increasing number of clusters in the within-cluster split, which contrasts with their behavior for accuracy. Standard decision trees' average sensitivity remained between 0.75 to 0.76 and multilevel decision trees' average sensitivity remained at 0.74, in the within-cluster split. This suggests that while decision trees may overfit or struggle with accuracy as J increases, they still maintain the ability to detect true positives within clusters as the number of clusters increases.

Overall, these results demonstrate that the effect of the number of clusters on model performance is context dependent. While larger J values generally enhance generalizability in the between-cluster split, their impact in the within-cluster split is limited for most models, except for decision trees which potentially experience overfitting.

Specificity, AUC, and F1 scores followed similar trends as accuracy and sensitivity (Figures A.7, A.8, A.9).

4.1.3.3 Effect of Individuals per Cluster: n_j

The number of individuals per cluster n_j was varied between 50 and 100, and the results demonstrate that this factor had minimal impact on model performance across both the within-cluster and between-cluster split conditions (Figure 4.8).

In the within-cluster split condition, outcome metrics remained largely consistent or showed only negligible changes as the number of individuals per cluster increased from 50 to 100 for most

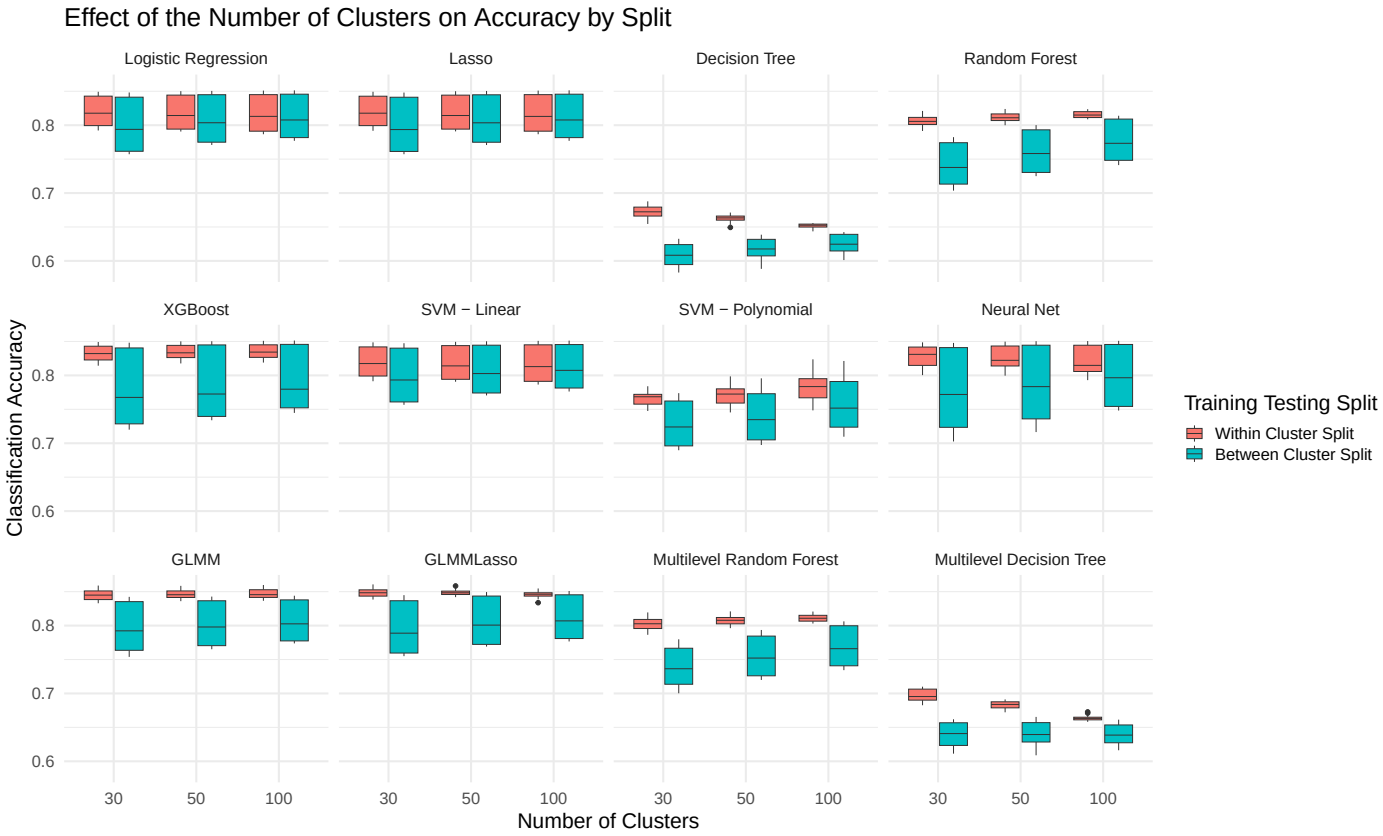

Figure 4.6: Classification accuracy by model, training testing split, and across levels of number of clusters, aggregated over residual level-2 variance, individuals per cluster, and data generating process.

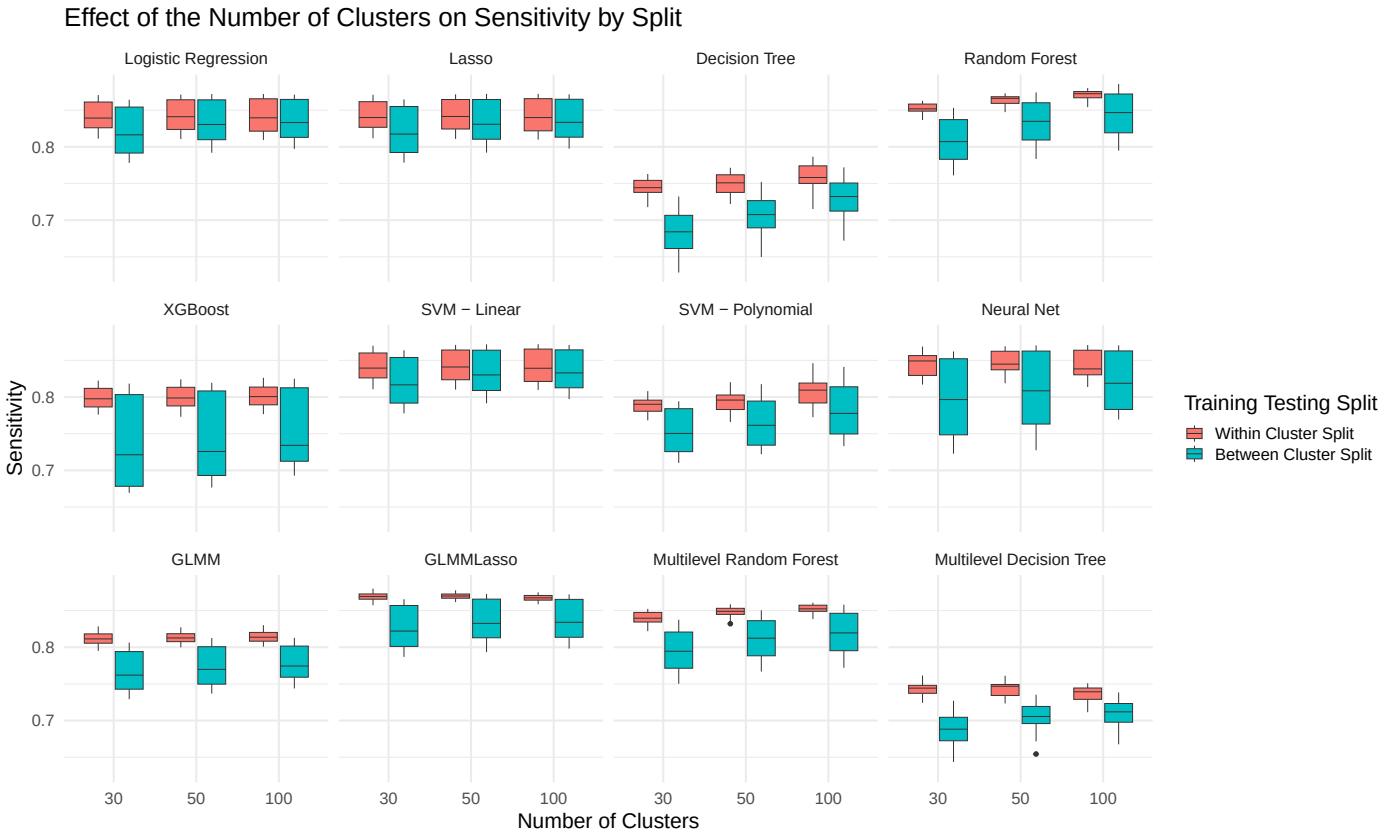

Figure 4.7: Test set sensitivity by model, training testing split, and across levels of number of clusters, aggregated over residual level-2 variance, individuals per cluster, and data generating process.

models. For instance, Logistic Regression's average classification accuracy remained at 82% as n_j increased from 50 to 100. Some models saw a modest 1% increase in outcome metrics, such as Random Forest which improved from 81% to 82% and XGBoost which increased from 83% to 84%.

For the between-cluster split condition, similar patterns were observed. Again, most models saw negligible changes in performance. GLMM maintained accuracy of 80% across both cluster size conditions. Neural Net, however, exhibited a slight decline in accuracy, decreasing from 79% to 78%.

Outcome metrics, including sensitivity, specificity, AUC, and F1 score, closely mirrored the trends observed for accuracy (Figure A.10, A.11, A.12, A.13). For example, XGBoost sensitivity improved slightly in the within-cluster split, increasing from 80% to 81%. GLMM specificity remained consistent in both split conditions, averaging at 83% for both $n_j = 50$ and $n_j = 100$. These additional metrics further support the conclusion that increasing n_j has minimal influence on performance across models.

Overall, the results indicate that the number of individuals per cluster does not substantially affect model performance under these simulation conditions.

4.1.3.4 Effect of Data-Generating Process: (DGP)

The data-generating process (DGP) varied across six conditions, each introducing different complexities to the model: Model A (Main Effects Random Intercept), Model B (Quadratic Nonlinear Random Intercept), Model C (Interaction Nonlinear Random Intercept), Model D (Exponential Nonlinear Random Intercept), Model E (Random Intercept and Random Slope), and

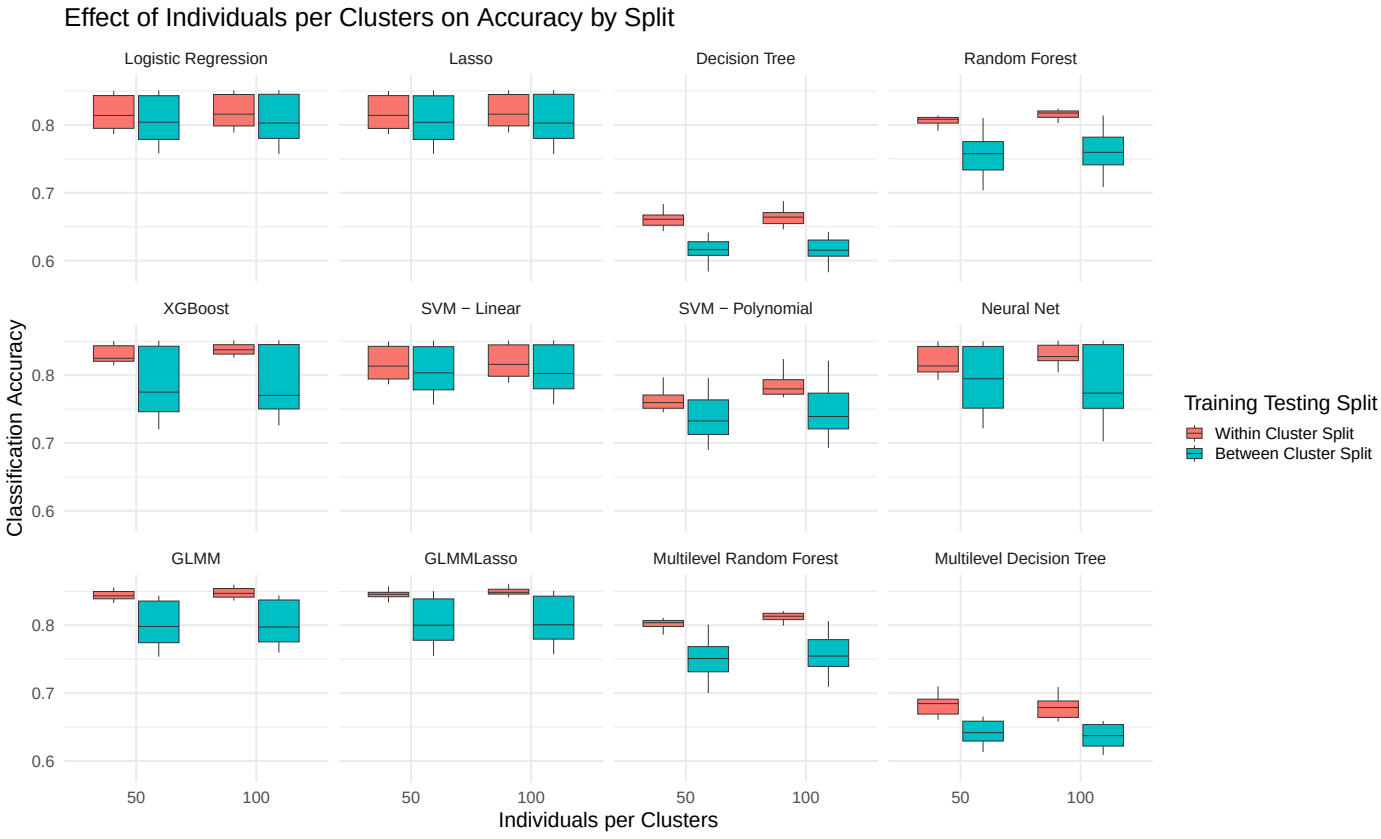

Figure 4.8: Classification accuracy by model, training testing split, and across levels of individuals per cluster. Results are aggregated across all other manipulated simulation factors, including residual level-2 variance, the number of clusters, and the data-generating process (DGP).

Model F (Random Intercept and Random Slope with Cross-Level Interaction). These conditions were designed to evaluate model performance under increasingly complex data structures. Model performance under these varying complexities of the data-generating processes (DGP) was aggregated over residual level-2 variance (σ^2_{u0}), the number of clusters (J), and the number of individuals per cluster (n_j).

As shown in Figure 4.9, which presents the average classification accuracy across all models and split conditions, accuracy remained relatively stable across all DGP levels. The figure highlights that most models were effective in capturing data patterns even as non-linear interactions and cross-level effects were introduced. In the within-cluster split, all models showed only a slight downward trend in accuracy as model complexity increased (Figure 4.9). For instance, Logistic Regression achieved an average classification accuracy ranging from 82% for Model A to 81% for Model F, showing its robustness even as data complexity increased. Similarly, GLMMLasso maintained high classification accuracy, ranging between 85% to 84% as the data generating model increased in complexity.

Unlike accuracy, sensitivity and specificity exhibited greater variability with increasing DGP complexity. Notably, Model D (Exponential Nonlinear Random Intercept) had the strongest impact on the sensitivity and specificity of standard decision trees and multilevel decision trees ((Figures 4.10, 4.11). The average sensitivity of standard decision trees dropped from 0.76 for Model A (Main Effects Random Intercept) to 0.73 for Model D (Figure 4.10, top row, third column). Similarly, multilevel decision trees experienced a decline in sensitivity, decreasing from 0.75 for Model A to 0.73 for Model D. All other models experienced a 1% dip in average sensitivity when comparing Model A to Model D, apart from XGBoost, which showed a 1% increase in average sensitivity (Model A: 0.80; Model D: 0.81).

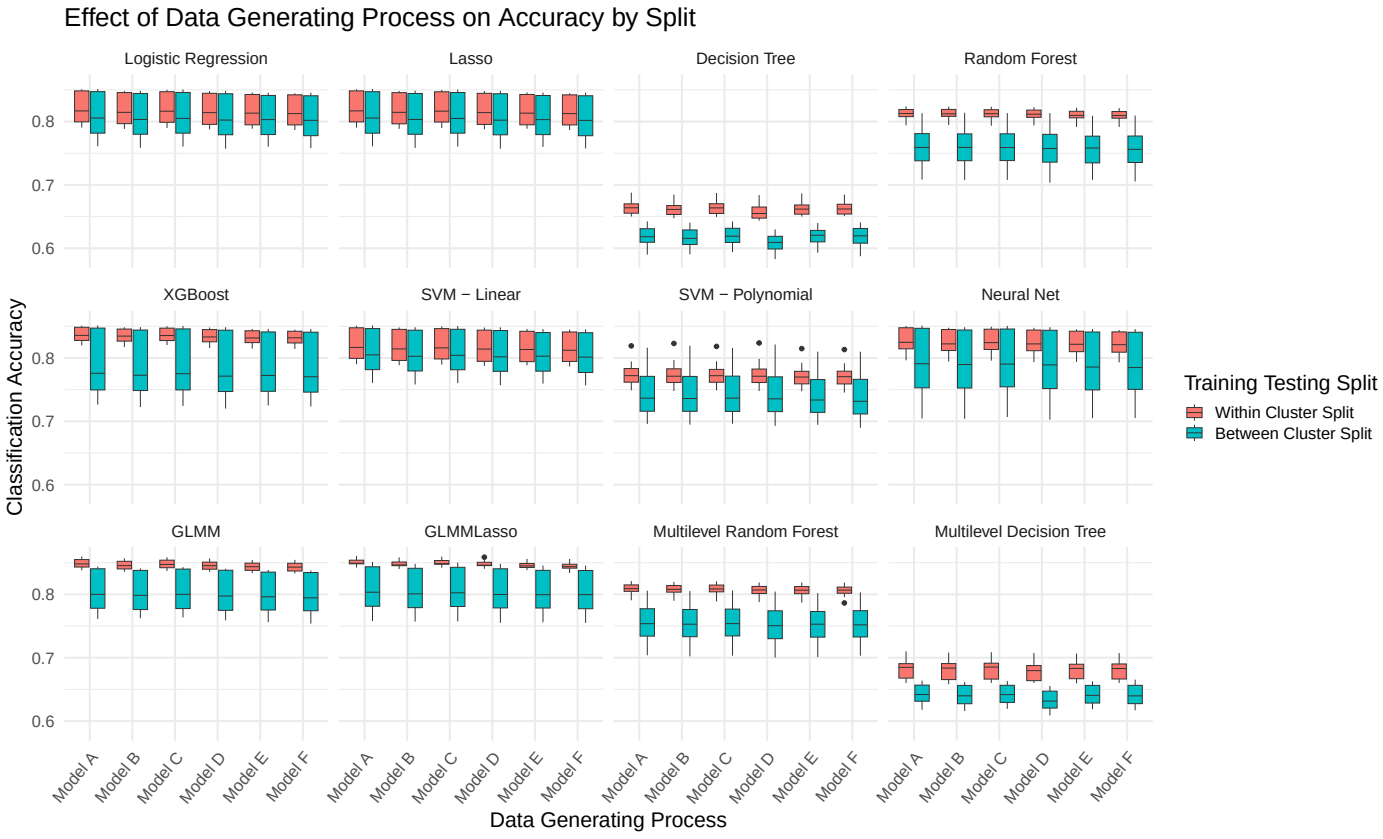

Figure 4.9: Classification accuracy by model, training testing split, and across data generating processes. Results are aggregated across all other manipulated simulation factors, including residual level-2 variance, the number of clusters, and individuals per cluster.

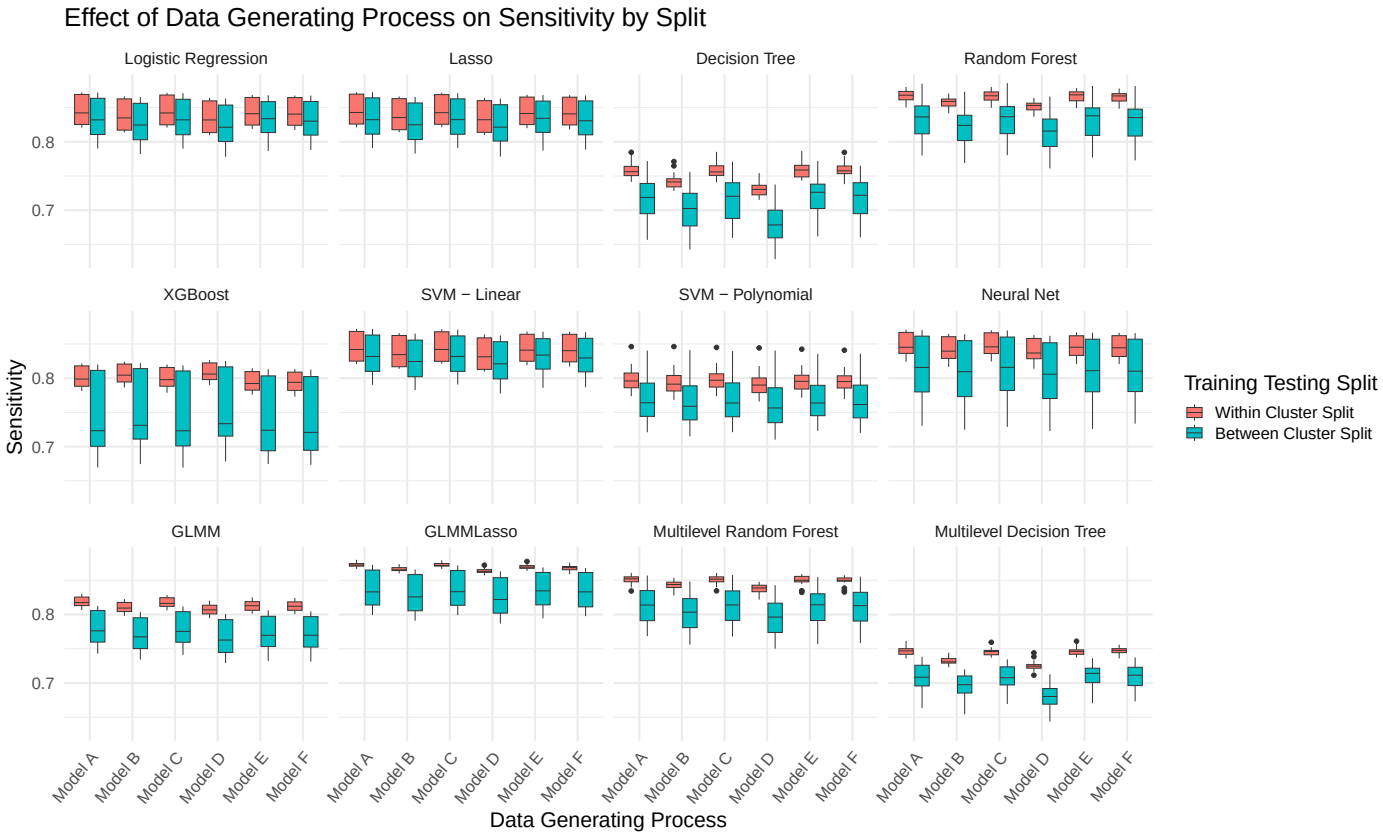

Figure 4.10: Test set sensitivity by model, training testing split, and across data generating processes. Results are aggregated across all other manipulated simulation factors, including residual level-2 variance, the number of clusters, and individuals per cluster.

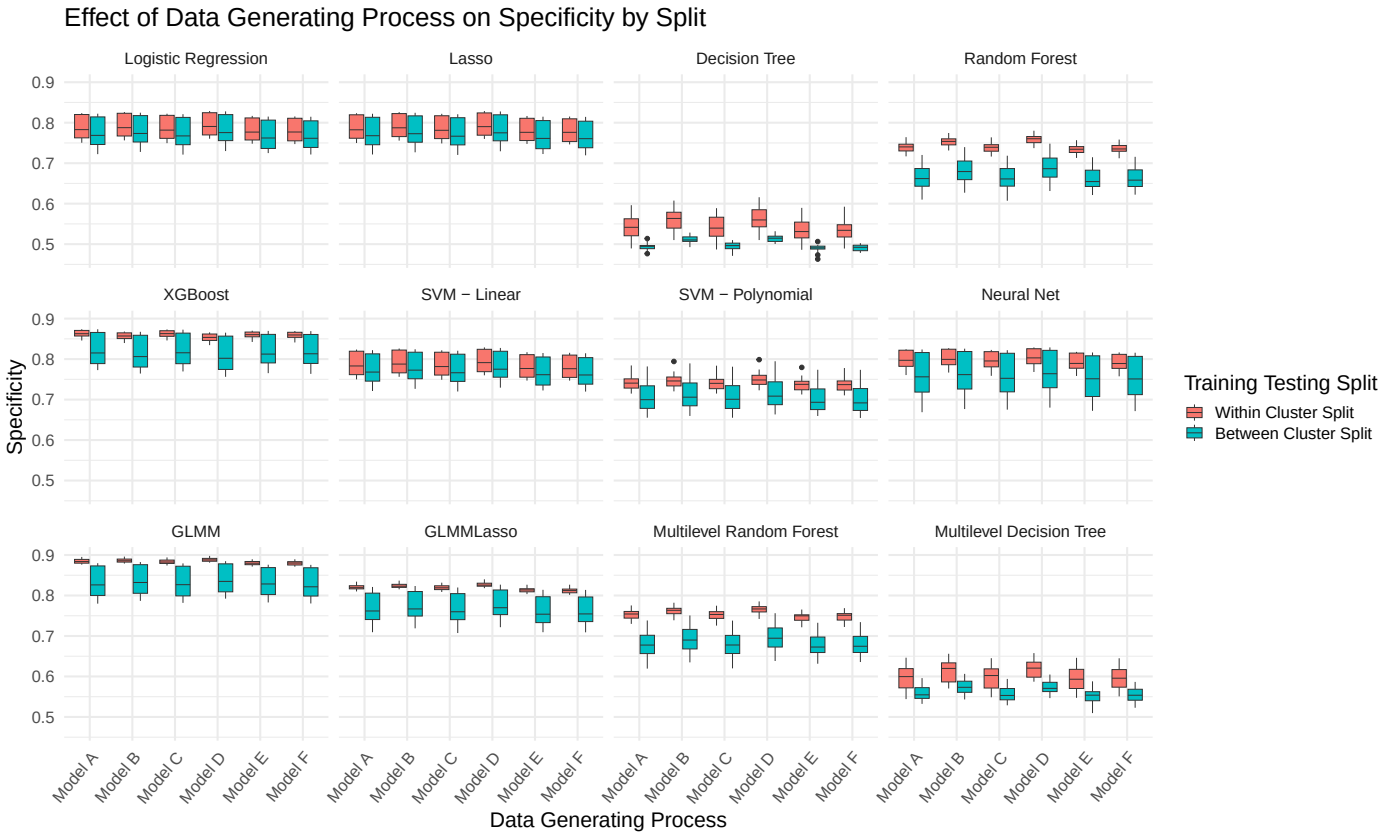

Figure 4.11: Test set specificity by model, training testing split, and across data generating processes. Results are aggregated across all other manipulated simulation factors, including residual level-2 variance, the number of clusters, and individuals per cluster.

4.1.4 Effects of Multiple Factors within the Training-Testing Split Context

The joint effects of the manipulated conditions were analyzed to identify any underlying trends in model performance that might have been obscured when aggregating over other factors. Overall, no significant interactions between factors were observed; the patterns evident in the results for individual factors generally persisted when considering the factors two at a time. Consistent with prior findings, residual level-2 variance (σ^2_{u0}) remained the most influential driver of variation in model performance.

Nevertheless, subtle trends emerged in specific contexts. For instance, in the between-cluster split condition, increasing the number of clusters (J) slightly mitigated the negative impact of higher residual level-2 variance on accuracy (Figure 4.12). When $\sigma^2_{u0} = 2$, the average accuracy of Logistic Regression improved from 0.79 (with $J = 30$) to 0.80 (with $J = 50$) and further to 0.81 (with $J = 100$). Similarly, when $\sigma^2_{u0} = 4$, the average accuracy increased from 0.76 (with $J = 30$) to 0.77 (with $J = 50$) to 0.78 (with $J = 100$). This relationship between residual level-2 variance and number of clusters was slightly stronger for Neural Net. For $\sigma^2_{u0} = 2$, the mean accuracy increased from 0.77 (with $J = 30$) to 0.78 (with $J = 50$) to 0.80 (with $J = 100$), reflecting a 3% improvement in average accuracy. When $\sigma^2_{u0} = 4$, the mean accuracy increased from 0.71 (with $J = 30$) to 0.73 (with $J = 50$) and even further to 0.75 (with $J = 100$). These improvements suggest that a higher number of clusters alleviates some of the challenges posed by increased residual level-2 variance σ^2_{u0}, leading to slightly better predictive performance.

Additionally, evaluating the number of clusters J and individuals per cluster n_j together offers insights into the impact of total sample size on model performance in both within-cluster

and between-cluster splits. As shown in Figure 4.13, most models display stable or slightly improved accuracy as either J or n_j increases. Standard Random Forest, Multilevel Random Forest, and SVM with Polynomial kernels appear to benefit the most from larger sample sizes. For example, SVM with Polynomial kernels demonstrates consistent improvements in accuracy as total sample size increases. In the within-cluster split condition, when $J = 30$, classification accuracy improves from 0.76 (with n_j=50) to 0.77 (with n_j=100). A slightly stronger trend is observed for $J = 50$, where classification accuracy improves from 0.76 (with n_j=50) to 0.78 (with n_j=100). Even stronger still, when $J = 100$, classification accuracy improves from 0.77 (with n_j=50) to 0.80 (with n_j=100), reflecting a 3% improvement among the $J = 100$ conditions and a 4% improvement when compared to the smallest sample size condition ($J = 30$ and n_j=50). In the between-cluster split condition, SVM with Polynomial kernels shows a similar pattern, just with slightly lower overall classification accuracy. At $J = 30$, classification accuracy increases from 0.73 (with $n_j = 50$) to 0.74 (with $n_j = 100$). For $J = 50$, classification accuracy improves from 0.74 (with $n_j = 50$) to 0.75 (with $n_j = 100$). Finally, at $J = 100$, classification accuracy increases from 0.75 (with $n_j = 50$) to 0.77 (with $n_j = 100$).

Finally, when evaluating the effects of the data-generating process (DGP) in the context of the three other factors (residual level-2 variance, number of clusters, and individuals per cluster), the complexity of the DGP consistently showed minimal impact on model performance (Figures A.17, A.18 , A.19). This reinforces its negligible effect across conditions, even when considered alongside these additional factors.

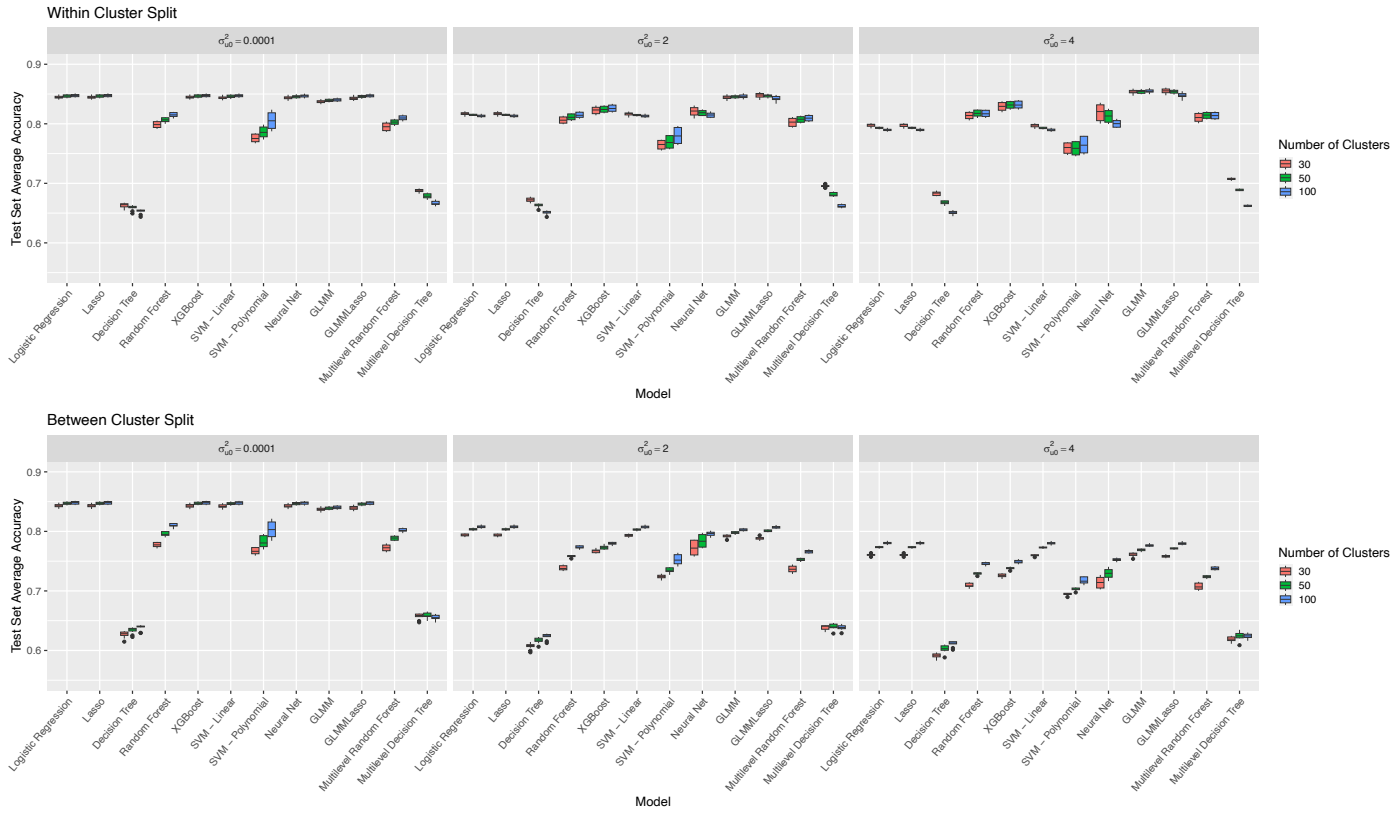

Figure 4.12: Average classification accuracy by model, split condition, residual level-2 variance (σ^2_{u0}), and the number of clusters (J). Each panel separates results by split condition (within-cluster vs between-cluster) and displays classification accuracy trends across models for varying levels of J and σ^2_{u0}. Results are aggregated across the number of individuals per cluster n_j and data generating processes.

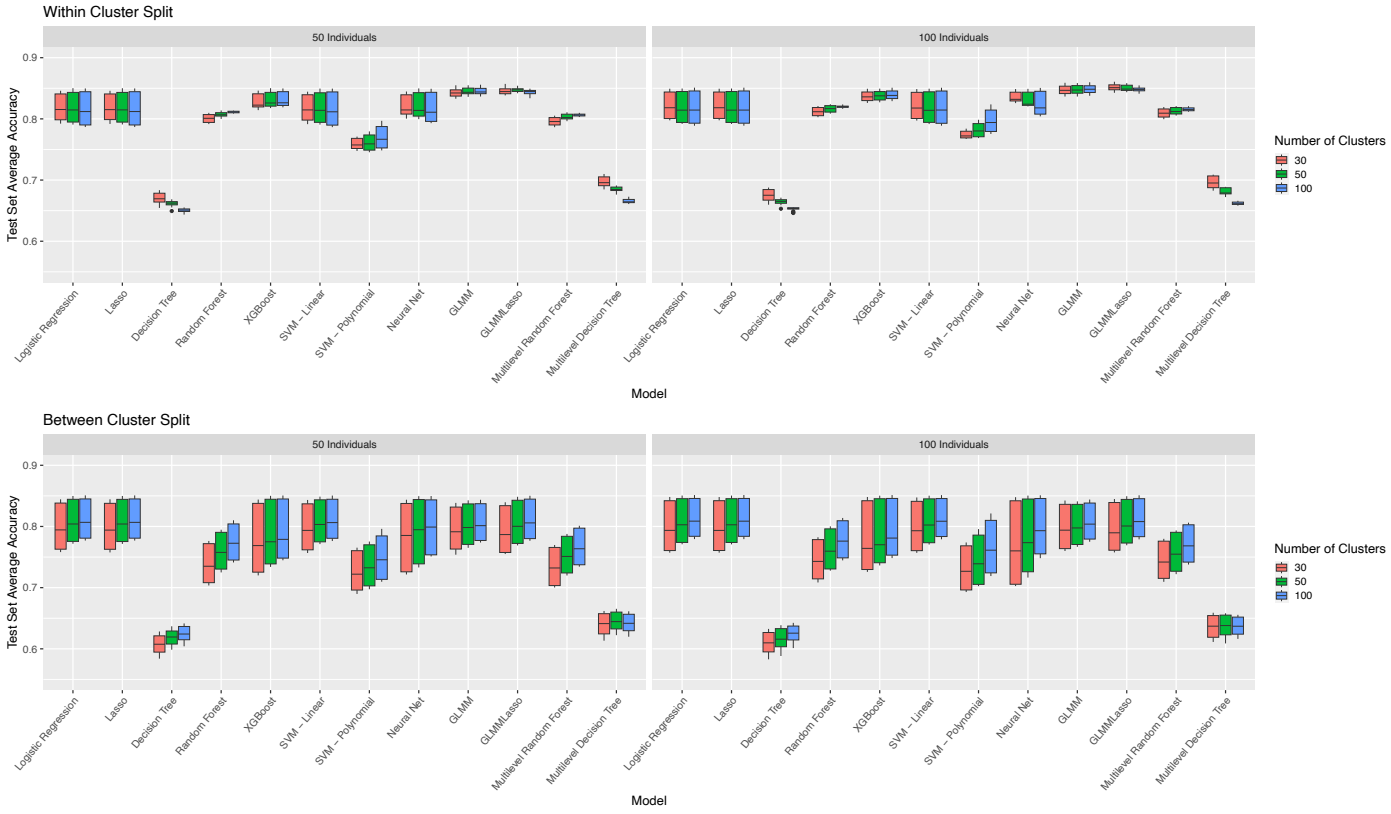

Figure 4.13: Average test set classification accuracy by model, split condition (within-cluster vs. between-cluster), number of clusters (J), and individuals per cluster (n_j). Results are aggregated across all residual level-2 variance levels (σ^2_{u0}) and data-generating processes to isolate the combined effects of J and n_j on model performance. Each panel highlights accuracy trends for varying J and n_j, allowing for a comparison of how total sample size influences classification accuracy across models and split conditions.

4.1.5 Effects of Cluster IDs as Fixed Effects in Standard Models

In addition to multilevel models, the effectiveness of including cluster identifiers (e.g., school IDs) as fixed effects in standard models was explored to determine if this approach could sufficiently capture cluster-level structure. The goal was to assess if this approach could substitute for a full multilevel model, providing insights into the possibility of modeling cluster-level effects directly in a fixed-effects context. This comparison was conducted within the within-cluster split condition, where clusters were represented in both the training and testing datasets.

Including cluster IDs as fixed effects had minimal impact on convergence rates for most models. For instance, Logistic Regression showed a slight decline in convergence, from 100% without cluster IDs to an average of 98% when cluster IDs were included, with a minimum convergence rate of 90%. However, this difference is unlikely to have practical implications. SVM with a Linear Kernel, however, failed to converge within the two-hour time limit under high sample size conditions (e.g., $J = 100$, $n_j = 100$), representing 18 of the total simulation conditions (Figure 4.14).

On average, standard models with cluster IDs as fixed effects outperformed their counterparts without cluster IDs and were often comparable to multilevel models across all outcome metrics (Figures 4.14, 4.15, 4.16, A.21, A.22). Logistic Regression, Lasso, XGBoost, and Neural Net achieved the highest average classification accuracy (0.85) when cluster IDs were included, matching the performance of top-performing multilevel models (GLMM and GLMM-Lasso).

In terms of sensitivity, models such as Logistic Regression, Lasso, SVM linear and Neural Net demonstrated the most improvement with the inclusion of cluster IDs, each of these mod-

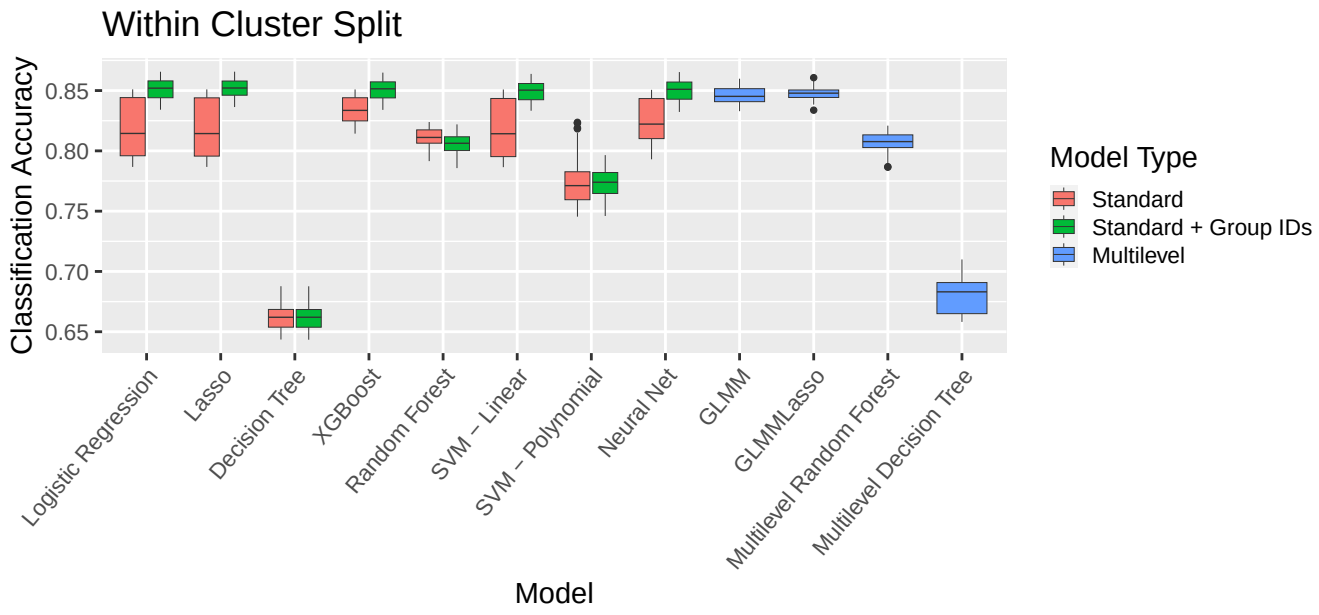

Figure 4.14: Classification accuracy by model and model type, including standard model, standard model including cluster IDs as fixed effects, and multilevel models. Results are for the within-cluster split condition and are aggregated across all other manipulated simulation factors, including residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

els improved from an average sensitivity of 0.84 to 0.87. This improvement positioned them alongside the previous top performer GLMMLasso which also had an average sensitivity of 0.87 (Figure 4.15).

Specificity results showed that GLMM exhibited the highest performance (0.83), followed closely by XGBoost (0.82) (Figure 4.16). Similar trends were observed for accuracy, AUC, and F1 score, further supporting the utility of incorporating cluster IDs as fixed effects (Figures 4.14, A.21, and A.22).

When breaking down the results further to examine the effects of including cluster IDs within the context of the other manipulated factors, standard models with cluster IDs behaved very similarly to the multilevel models. For example, as residual level-2 variance (σ^2_{u0}) increased,

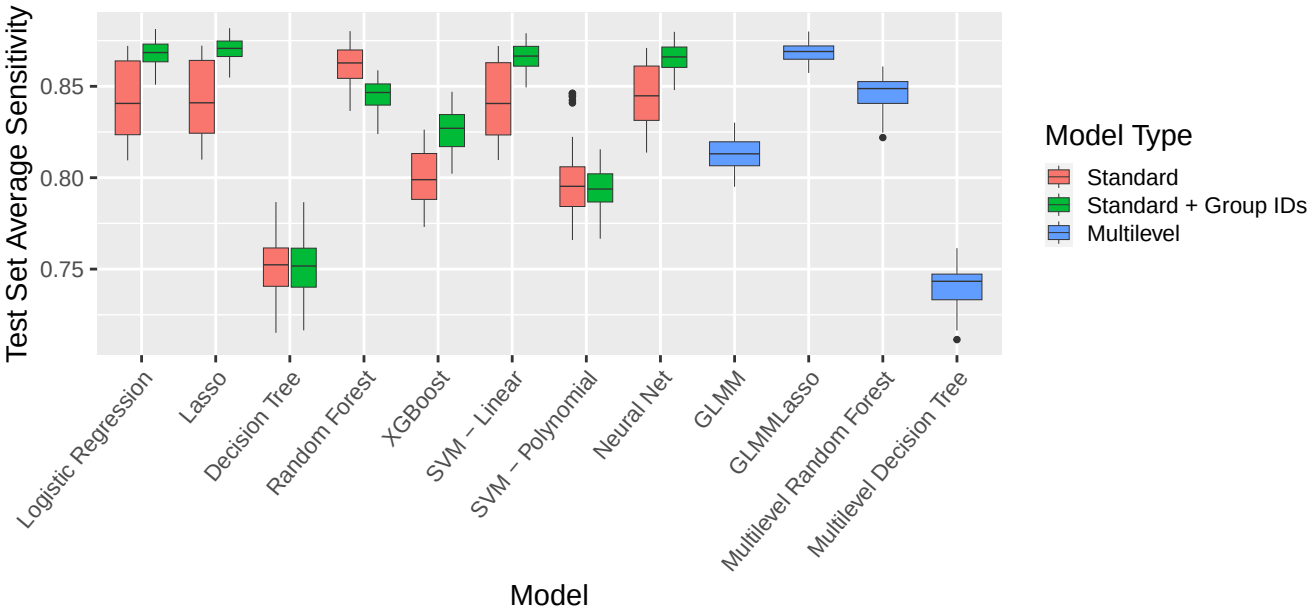

Figure 4.15: Test set sensitivity rates by model and model type, including standard model, standard model including cluster IDs as fixed effects, and multilevel models. Results are for the within-cluster split condition and are aggregated across all other manipulated simulation factors, including residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

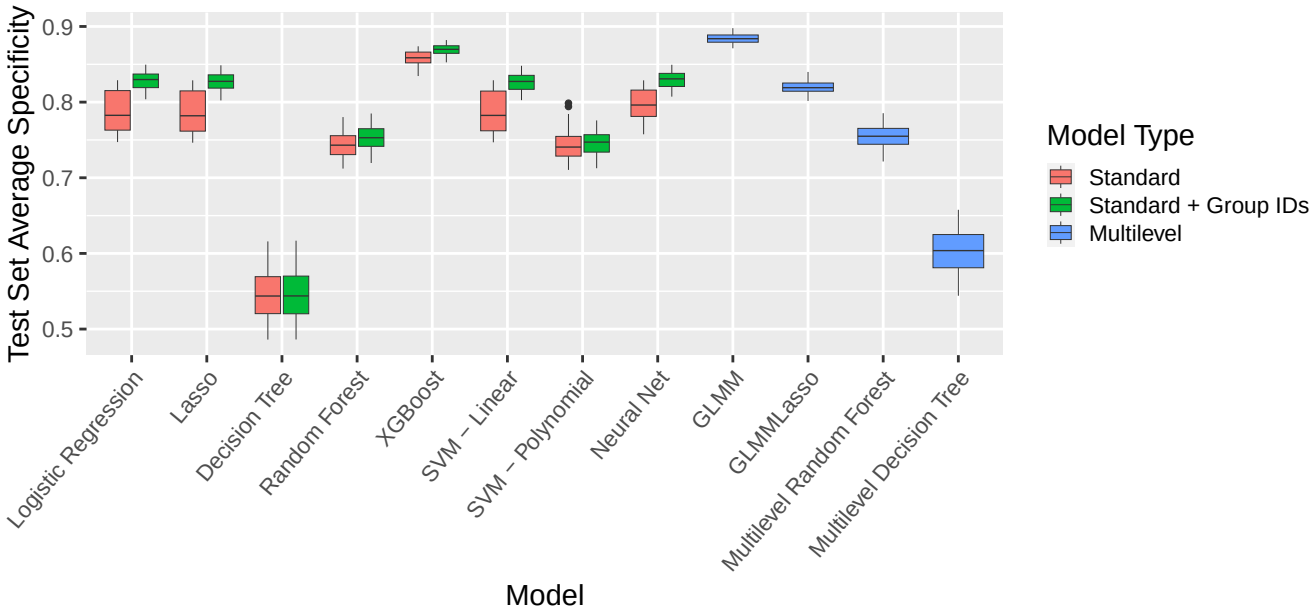

Figure 4.16: Test set specificity rates by model and model type, including standard model, standard model including cluster IDs as fixed effects, and multilevel models. Results are for the within-cluster split condition and are aggregated across all other manipulated simulation factors, including residual level-2 variance, number of clusters, number of individuals per cluster, and data-generating process.

standard models with cluster IDs showed improved accuracy, aligning closely with trends observed in multilevel models (Figure 4.17). Specifically, Lasso with cluster IDs as fixed effects achieved an average classification accuracy of 0.84, 0.85, and 0.86 for $\sigma^2_{u0} = 0.0001, 2,$ and 4, respectively. By comparison, GLMMLasso, Lasso's multilevel counterpart, exhibited an average classification accuracy of 0.85 across all levels of σ^2_{u0}. While GLMMLasso demonstrated more stable performance regardless of residual level-2 variance, these results suggest that including cluster IDs as fixed effects enables standard models like Lasso to effectively account for cluster-level variance. Notably, at the highest level of residual level-2 variance ($\sigma^2_{u0} = 4$), Lasso outperformed its multilevel counterpart, achieving a slightly higher average classification accuracy.

Additional plots exploring the joint effects of other manipulated factors and model type can be found in the Appendix (Figures A.23, A.24, and A.25).

4.1.5.1 Best-Performing Models and Implications

Among the standard models with cluster IDs, Lasso and XGBoost consistently demonstrated strong performance across accuracy, sensitivity, and specificity, making them suitable choices for capturing cluster-level structure in this context. For example, Lasso achieved an average classification accuracy of 0.86 at $\sigma^2_{u0} = 4$, demonstrating its ability to maintain high performance even at the highest level of residual level-2 variance. It also recorded the highest average sensitivity rates, with 0.87 at $\sigma^2_{u0} = 2$ (on par with GLMMLasso) and 0.88 at $\sigma^2_{u0} = 4$, slightly surpassing GLMMLasso's 0.87. XGBoost with cluster IDs as fixed effects, on the other hand, excelled in specificity, closely following the top performance of GLMM. At $\sigma^2_{u0} = 0.0001$,

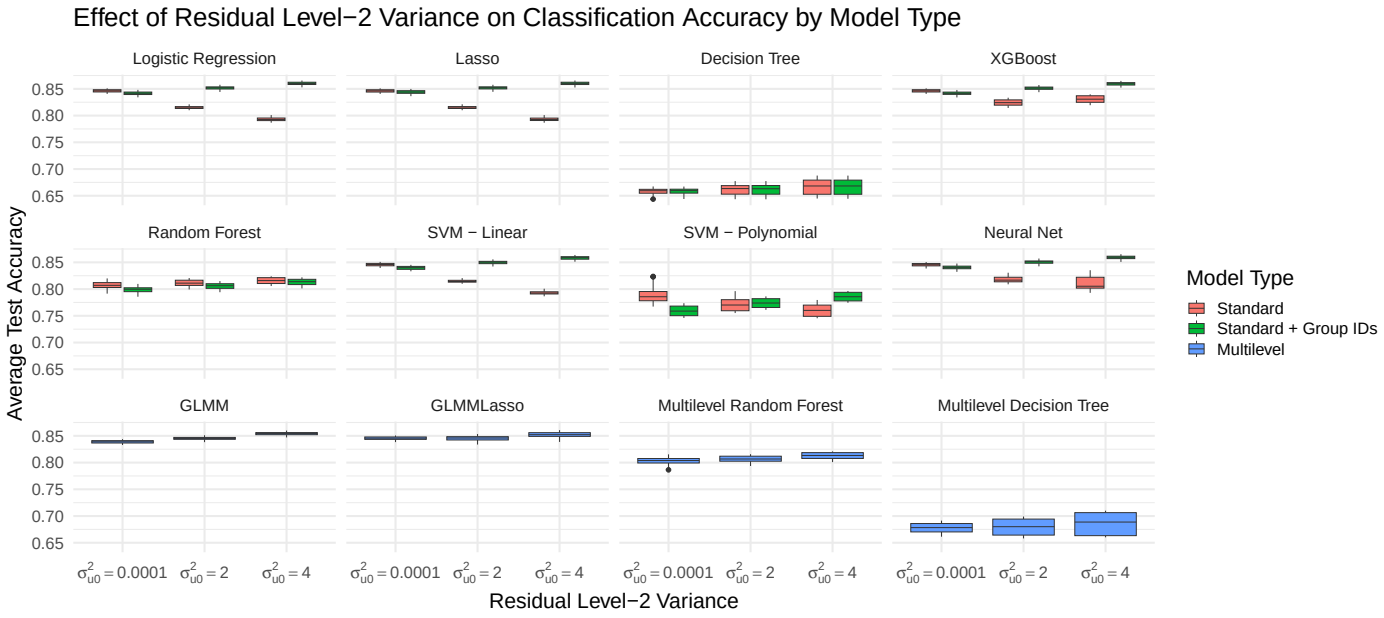

Figure 4.17: Test set classification accuracy by model, residual level-2 variance, and model type, including standard model, standard model including cluster IDs as fixed effects, and multilevel models. Results are for the within-cluster split condition and are aggregated across all other manipulated simulation factors, including number of clusters, number of individuals per cluster, and data-generating process.

XGBoost achieved an average specificity of 0.86, compared to GLMM's 0.88. For $\sigma^2_{u0} = 2$ XGBoost's average specificity was 0.87, slightly below GLMM's 0.88. At $\sigma^2_{u0} = 4$, XGBoost reached 0.88, closely matching GLMM's highest value of 0.89. Additionally, XGBoost maintained a strong average accuracy ranging from 0.84 to 0.86 across the levels of residual level-2 variance, underscoring its robust performance.

Along with Logistic Regression and Neural Net, both of which maintained average accuracy ranging from 0.84 to 0.86 across the levels of residual level-2 variance, these models demonstrate the potential for standard models with cluster IDs to perform on par with or even surpass multilevel models in many scenarios.

The main takeaway from this within-cluster analysis is that standard models which include cluster IDs as predictors offer a viable alternative to multilevel models when predictions are needed for new individuals in clusters represented in the training set. These models seem to be capable of capturing cluster-level effects, even under conditions of higher residual level-2 variance (σ^2_{u0}) that initially challenged standard models, achieving performance closely aligned with multilevel models.

4.1.6 Summary of Simulation Study Results

This simulation study examined the predictive performance of standard and multilevel machine learning algorithms under various hierarchical data conditions. The goal was to evaluate how different factors, including training-testing splits, residual level-2 variance (σ^2_{u0}), the number of clusters (J), the number of individuals per cluster (n_j), and the complexity of the data-generating process, influence model outcomes.

Multilevel models like GLMM and GLMMLasso generally provided robust and stable predictions, particularly under conditions of high σ^2_{u0}, where their ability to model both individual- and cluster-level dependencies resulted in slightly higher average performance and greater stability. In contrast, more complex multilevel algorithms such as BiMM Forest and M-CART offered no clear practical advantages over these simpler multilevel approaches.

Standard models, particularly those able to incorporate cluster IDs as fixed effects, performed competitively in many scenarios. In within-cluster splits conditions, where clusters were consistent across training and testing, standard models which included cluster IDs as fixed effects often aligned closely with multilevel models. For example, Logistic Regression, Lasso, and XGBoost achieved comparable levels of accuracy, sensitivity and specificity. Lasso, in particular, demonstrated strong performance under high σ^2_{u0} achieving an average classification accuracy of 0.86 and sensitivity of 0.88 at $\sigma^2_{u0} = 4$. Additionally, when the effects of hierarchical structure were less pronounced (e.g., $\sigma^2_{u0} = 0.0001$), standard algorithms performed similarly to multilevel algorithms across both split conditions.

In the between-cluster split condition, model performance declined across the board due to the absence of cluster-level representation in the training data. As σ^2_{u0} increased, neither standard nor multilevel models consistently outperformed the other. Increasing the number of clusters (J) appeared to help mitigate the negative impact of higher residual level-2 variance on accuracy slightly. For instance, Neural Nets and Logistic Regression experienced smaller performance declines under high (σ^2_{u0}) when more clusters were present in the data.

Larger total sample sizes, achieved through increases in J or n_j, also tended to improve model performance slightly across most algorithms. This effect was particularly pronounced for ensemble methods such as XGBoost and Random Forest. Finally, the complexity of the DGP had

little impact on model outcomes, suggesting that these models were robust to variations in how the data were generated.

4.1.7 Recommendations for Education Researchers

This simulation study provided several insights that can guide education researchers in selecting appropriate predictive models for hierarchical data, depending on their datasets and research objectives.

Leverage standard models for within-cluster predictions: When predictions are made within the same clusters represented in the training set (e.g., predicting outcomes for new students within schools already observed), standard models incorporating cluster IDs as fixed effects (e.g., Logistic Regression, Lasso, XGBoost) offer a robust and computationally efficient alternative to multilevel models. These models performed comparably to multilevel models in this context and provide flexibility and easy implementation for researchers.

Consider limitations of between-cluster predictions: The between-cluster split condition, where predictions are made for entirely new clusters (e.g., schools not observed in the training set), posed challenges for both standard and multilevel models. As residual level-2 variance (σ^2_{u0}) increased, neither standard nor multilevel models consistently outperformed the other. In such cases, researchers should cautiously interpret predictions under these conditions and consider collecting additional level-2 predictors to better account for cluster-level variability or increasing the number of clusters in their datasets.

Use model simplicity to your advantage: When high-quality level-2 predictors reduce residual level-2 variance (e.g., $\sigma^2_{u0} \approx 0.0001$), standard models achieved performance compara-

ble to multilevel models. These models offer a simpler and computationally less intensive alternative for researchers, especially in contexts where interpretability and ease of implementation are priorities.

Do not overemphasize data complexity: The complexity of the data-generating process (DGP) had little impact on model performance across conditions, suggesting that researchers can apply these models for prediction tasks without much concern for variations in underlying DGP complexity.

Chapter 5: Empirical Illustration

This chapter presents an empirical illustration conducted to evaluate the performance of standard and multilevel machine learning algorithms in predicting STEM degree completion using data from MLDS. These analyses utilized two datasets: a full sample of students and an imbalanced sample focusing on Black students receiving Pell Grants. The study assessed the impact of train-test splitting strategies (within-cluster and between-cluster splits) and examined the role of including cluster identifiers as fixed effects in standard models. Additionally, this study investigated how class imbalance influenced model performance and emphasized specificity and sensitivity as key evaluation metrics.

The findings of this study have potential implications for educational researchers and policymakers aiming to improve STEM persistence. By identifying models that effectively predict STEM degree completion, especially for underrepresented clusters, this study highlights approaches that can guide targeted interventions. Models like Neural Nets and XGBoost, which demonstrated strong performance across metrics for imbalanced datasets, can help policymakers prioritize at-risk students for academic or financial support.

5.1 Sample Descriptions

5.1.1 Full Sample (N = 4,241)

The full sample included 4,241 students who declared STEM degrees in the fall semester of their freshman year at twelve Maryland Universities. To minimize the impact of external disruptions such as the COVID-19 pandemic, the dataset included students who were first-time freshmen in Fall 2014. Among these students, 57.3% (2,432 students) successfully earned a STEM degree within six years, while 42.7% (1,809 students) did not. The universities included in the full sample spanned a diverse range of public and private institutions in Maryland. The University of Maryland College Park accounted for the largest cluster, with 1,592 STEM students, followed by the University of Maryland Baltimore County with 1,027 students. Smaller institutions, such as Frostburg State University and Capitol College, contributed 77 and 31 students, respectively, illustrating substantial variation in cluster sizes across the twelve universities (Table 5.1).

Individual-level characteristics indicated an average SAT/ACT score of 1219 (SD = 193), a freshman fall cumulative GPA of 3.0 (SD = 0.9), and an average of 13.7 (SD = 3.3) credits earned in their first semester. The majority of students were Maryland residents (78.7%), and the racial composition included 53.4% White, 21.6% Black, 17.9% Asian, and 7.1% Other. The gender breakdown was skewed toward males (59.0%), with females making up only 41.0% of the sample. Regarding financial aid, 74.9% filed a FAFSA, and 22.3% of the students received Pell Grants. These descriptive statistics, summarized in Table 5.2, highlight the diversity of the full sample across demographic and institutional factors. An unconditional random intercept model

determined the ICC of this dataset as 0.11, reflecting a moderate degree of within-university variation in STEM persistence.

Table 5.1: Universities and number of STEM students in full and imbalanced samples

University	**Full Sample (N = 4,241)**	**Imbalanced Sample (N = 434)**
University of Maryland College Park	1,592	44
University of Maryland Baltimore County	1,027	41
Towson University	499	55
Salisbury University	240	-
Frostburg State University	77	16
Stevenson University	130	32
Loyola University Maryland	202	-
Bowie State University	79	31
Coppin State University	32	22
Morgan State University	207	108
University of Maryland Eastern Shore	183	79
Capitol College	31	6

5.1.2 Imbalanced Sample (N = 434)

The imbalanced sample focused on a subset of 434 Black students who received Pell Grants at 10 Maryland Universities. Among them, only 15.0% (65 students) completed a STEM degree within four years, while 85.0% (369 students) did not. In this sample, Morgan State University represented the largest cluster with 108 students, while institutions such as Frostburg State University and Capitol College contributed only 16 and 6 students, respectively (Table 5.1).

Compared to the full sample, this group had lower average SAT/ACT scores (979.2, SD = 153.2), a lower freshman Fall GPA (2.6, SD = 0.9), and fewer credits earned in their first semester (11.8, SD = 3.9). The gender distribution was nearly equal, with males comprising 49.5% of the sample and females 50.5%. Most students (83.2%) were Maryland residents, and nearly all attended public universities (91.2%).

Institutional characteristics also differed from the full sample, with only 32.0% of students in the imbalanced sample attending large institutions. A greater proportion attended medium-sized (35.7%) or small (32.0%) universities. These distinctions, summarized in Table 5.2, underscore the unique characteristics of the imbalanced sample. An unconditional random intercept model calculated the ICC of the imbalanced sample dataset as 0.10, which was similar to the full dataset.

Table 5.2: Descriptive statistics for full and imbalanced samples

Variable	**Full Sample (N=4,241)**	**Imbalanced Sample (N=434)**
STEM Degree Completion (Yes)	57.3%	15.0%
STEM Degree Completion (No)	42.7%	85.0%
Maximum SAT/ACT Score (Mean $\pm$ SD)	1219 $\pm$ 193	979.2 $\pm$ 153.2
Freshman Fall Cumulative GPA (Mean $\pm$ SD)	3.0 $\pm$ 0.9	2.6 $\pm$ 0.9
Freshman Fall Credits Earned (Mean $\pm$ SD)	13.7 $\pm$ 3.3	11.8 $\pm$ 3.9
Maryland Resident (Yes)	78.7%	83.2%
Maryland Resident (No)	21.3%	16.8%
Race - White	53.4%	0%
Race - Black	21.6%	100.0%
Race - Asian	17.9%	0%
Race - Other	7.1%	0%
Gender - Male	59.0%	49.5%
Gender - Female	41.0%	50.5%
Filed a FAFSA (Yes)	74.9%	100%
Filed a FAFSA (No)	25.1%	0%
Received Pell Grant (Yes)	22.3%	100.0%
Received Pell Grant (No)	77.7%	0%
College Average Freshman Fall GPA (Mean $\pm$ SD)	3.0 $\pm$ 0.2	2.7 $\pm$ 0.3
College Average Freshman Fall Credits Earned (Mean $\pm$ SD)	13.5 $\pm$ 1.0	12.2 $\pm$ 1.2
Public University (Yes)	92.7%	91.2%
Public University (No)	7.3%	8.8%
College Size - Large	73.5%	32.0%
College Size - Medium	17.6%	35.7%
College Size - Small	8.9%	32.0%

5.2 Predicting STEM Persistence: Full Sample Analyses

5.2.1 Train-Test Split Within Clusters

When training and testing sets were split within clusters, model performance across evaluation metrics varied (Figure 5.1). Logistic Regression, Lasso, and GLMMLasso achieved comparable classification accuracies (0.72-0.73) and high sensitivity (0.84), making them particularly effective in identifying students likely to complete their STEM degree.

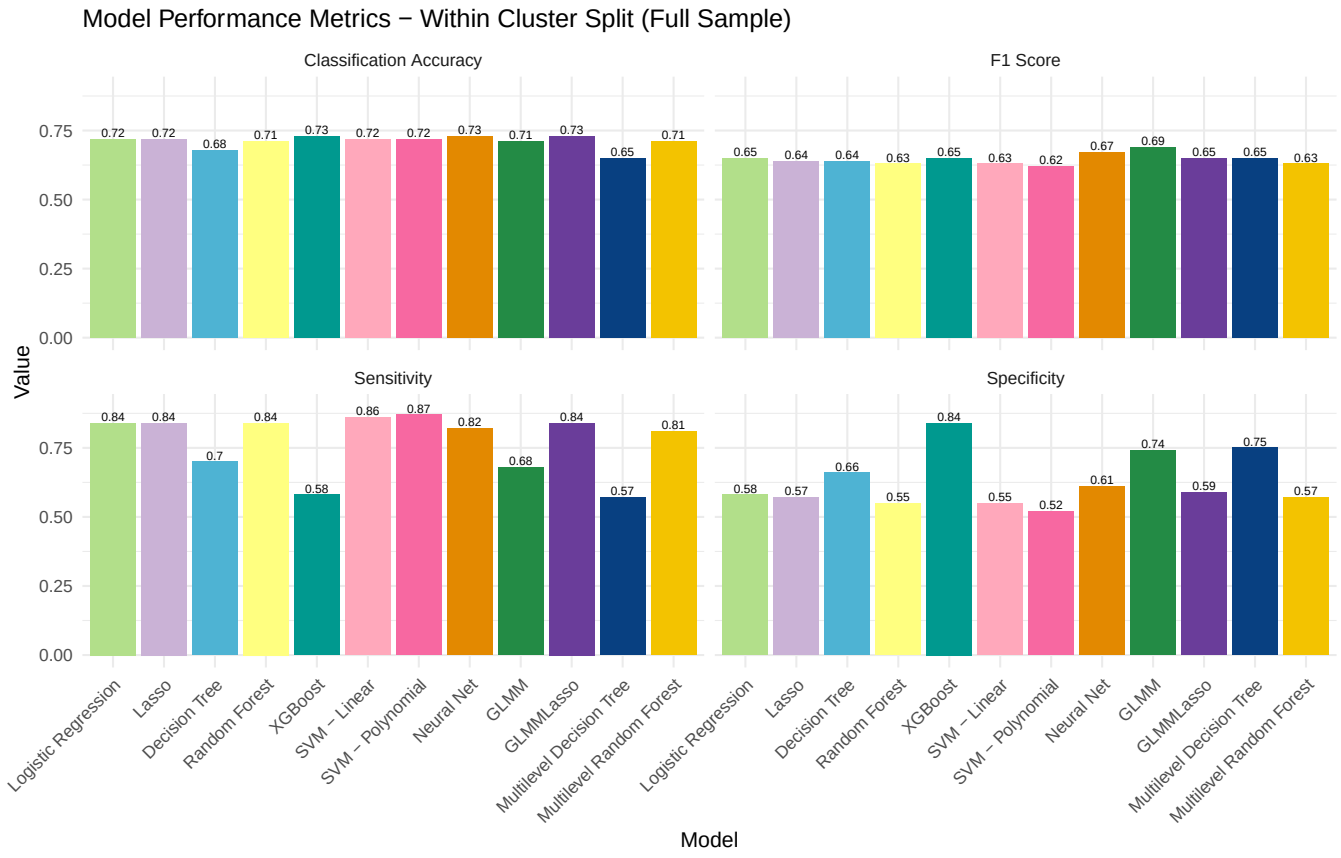

Figure 5.1: Performance metrics for standard and multilevel models, evaluated on their ability to predict STEM persistence using the full sample with a within-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

SVM models demonstrated the strongest sensitivity (0.86-0.87), indicating their effectiveness in identifying STEM degree completers. However, this strength came at the expense of specificity, which was lower compared to other models. Despite good overall classification accuracy relative to other models (0.72), this trade-off suggests that SVM models may overpredict completers, which could limit their utility in scenarios where specificity is critical.

XGBoost stood out for its high specificity (0.84), highlighting its strength in identifying students who did not graduate with a STEM degree. While its classification accuracy (0.73) was comparable to simpler models like Logistic Regression, its ability to pinpoint at-risk students underscores its potential utility for designing targeted interventions aimed at improving retention.

GLMM delivered a balanced performance across metrics, with sensitivity of 0.68 and specificity 0.74, resulting in the highest F1 score (0.69) and one of the highest AUC scores (0.71). However, since neither its sensitivity nor specificity was the highest, other models might be more desirable for specific scenarios, such as XGBoost for identifying non-completers or Logistic Regression and SVM model for identifying completers. Multilevel random forests did not show clear advantages over standard algorithms, while decision trees (both standard and multilevel) performed poorly, with consistently low metrics across the board. Their poor classification accuracy (as low as 0.65) indicate that they are not suitable for this context.

Adding cluster IDs as fixed effects to standard models led to slightly improvements in performance across multiple metrics under this train-test split condition (Figure 5.2). Including cluster IDs likely enhanced the models by incorporating additional context about institutional characteristics influencing STEM degree completion, which helped improve predictions for both students who completed and those who did not complete their STEM degrees. For example, Lasso with cluster IDs retained its sensitivity (0.84) while slightly improving its classification

accuracy (0.73) and specificity (0.58).

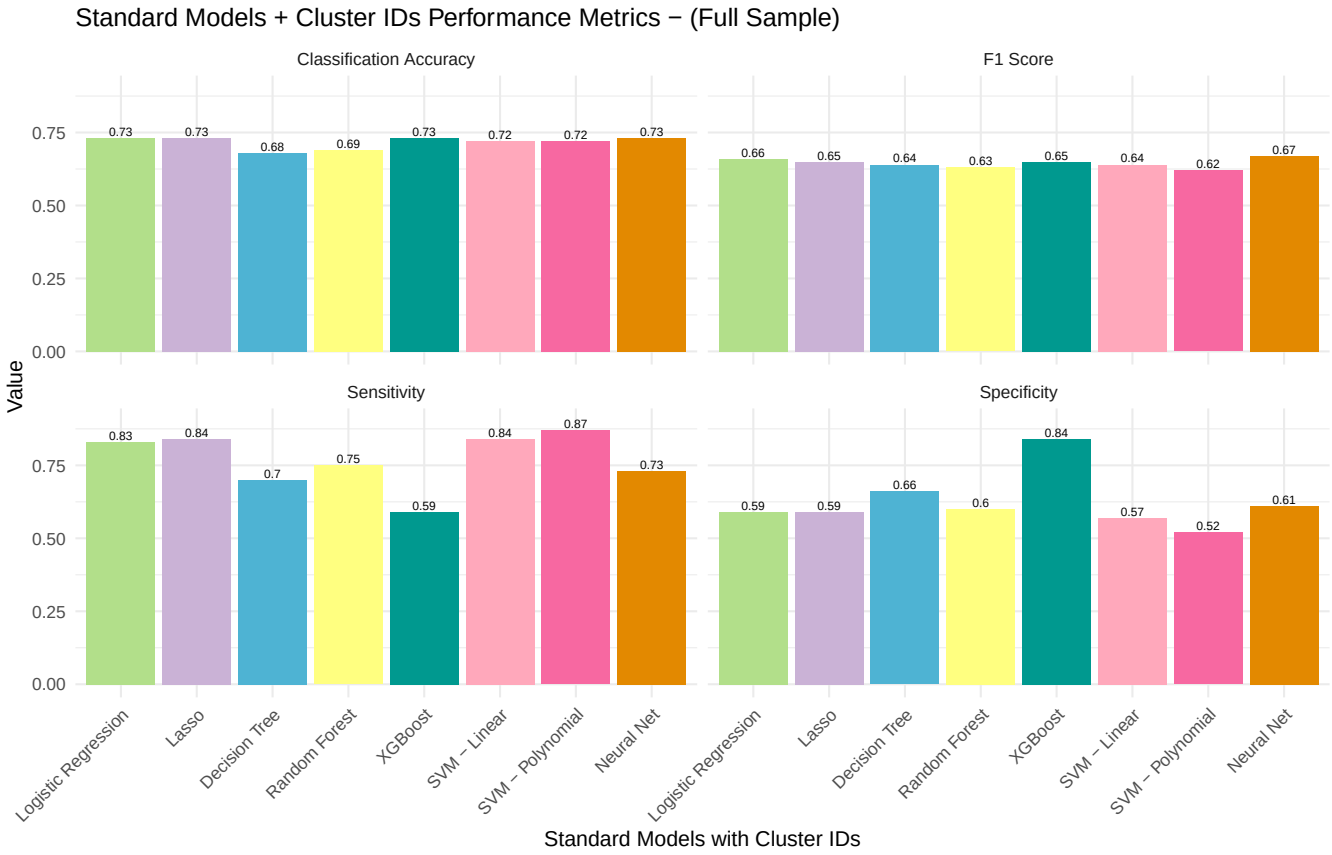

Figure 5.2: Performance metrics for standard models with cluster IDs as fixed effects, evaluated on their ability to predict STEM persistence using the full sample with a within-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

Overall, the within-cluster split results suggest that standard models, such as Logistic Regression, SVMs and XGBoost (with or without cluster IDs), can perform as well as or better than multilevel models. XGBoost's strength in identifying students who did not persist in STEM fields could be especially valuable for developing targeted interventions to support at-risk students. By contrast, models like GLMM may be preferred when a balanced approach is needed. These results provide a nuanced understanding of how model selection should align with specific goals,

such as maximizing sensitivity or specificity.

5.2.2 Train-Test Split Between Clusters

In the between-cluster split scenario, model performance declined across most metrics, reflecting the inherent difficulty of generalizing to unseen clusters (Figure 5.3). Logistic Regression, Lasso, and Neural Nets maintained moderate classification accuracies (0.67), but their ability to identify students who did not graduate with a STEM degree was limited, as evidenced by low specificities (0.56-0.57). Neural Nets achieved the highest AUC (0.67) among the standard models, but its less than ideal F1 score of 0.60 suggested only marginal adaptability to the more complex task of generalizing across clusters.

Decision trees and XGBoost demonstrated slightly lower classification accuracy (0.64-0.65), but maintained balanced specificity and sensitivity, offering more consistent performance across metrics. Multilevel models demonstrated stronger performance than standard models in some metrics. GLMM stood out with the highest classification accuracy (0.67), specificity (0.75), AUC (0.67), and F1 score (0.67), highlighting its strengths under these conditions. However, multilevel models also faced challenges with sensitivity which limited their ability to reliably identify STEM degree completers. Multilevel random forests performed well in identifying non-completers (specificity 0.71) but exhibited lower sensitivity (0.60), suggesting limitations in their overall utility.

Overall, no single model consistently outperformed others across all metrics in the between-cluster split condition. GLMM offered the best performance across the most evaluation metrics; however, neural nets closely followed in effectiveness. These results align with findings from the

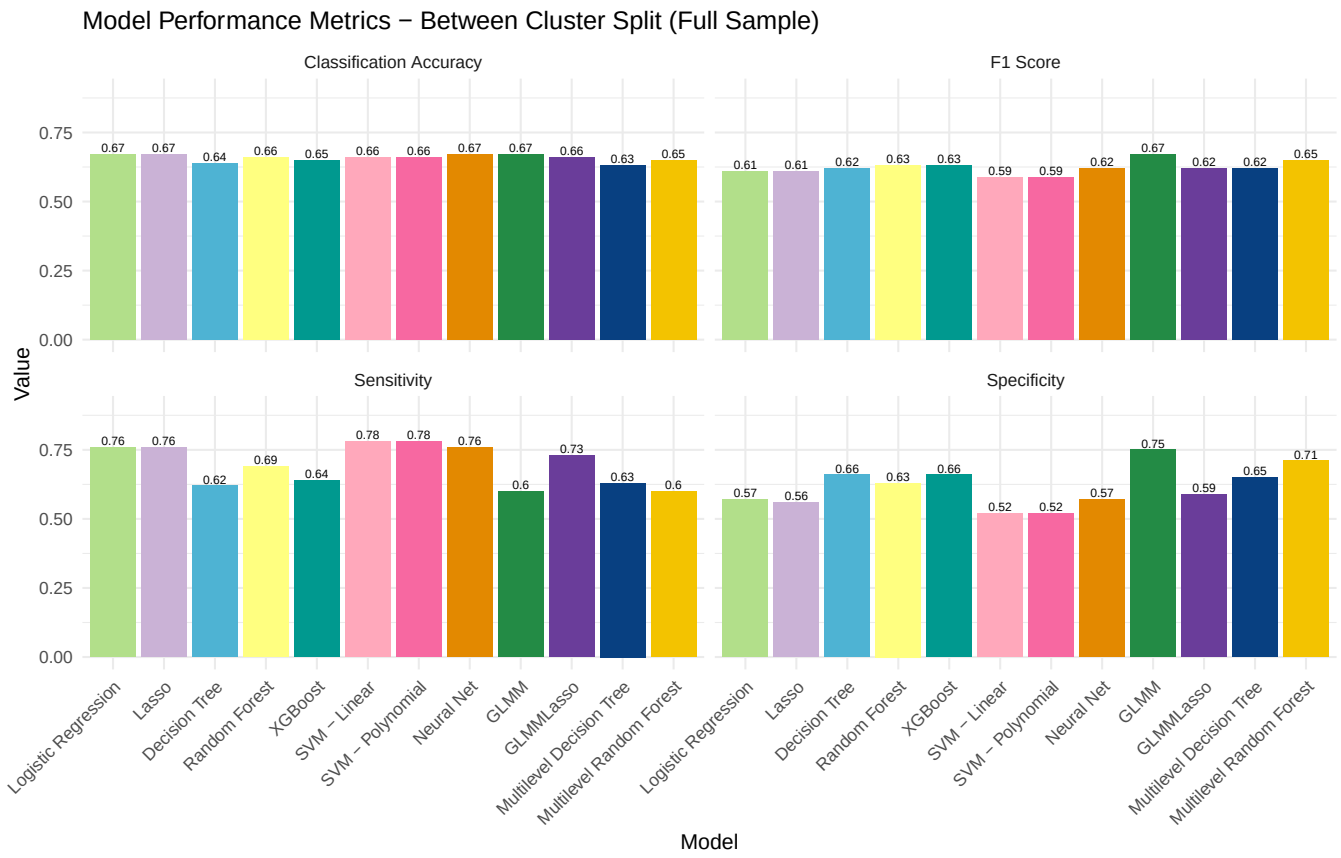

Figure 5.3: Performance metrics for standard and multilevel models, evaluated on their ability to predict STEM persistence using the full sample with a between-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

simulation study, where performance similarly declined across all models in the between-cluster split condition, and no model consistently excelled across all metrics. The general decline in performance observed here underscores the challenges of generalizing predictions across clusters in hierarchical data

5.3 Predicting STEM Persistence: Imbalanced Sample Analyses

For the imbalanced dataset, the ROC curve was used to establish the optimal threshold value for each model to ensure that predictions were better tailored to the skewed class distribution. These analyses demonstrate how models handle the smaller class, which in these analyses consist of Black students who received a Pell Grant and graduated with a STEM degree in four years.

5.3.1 Train-Test Split Within Clusters

The within-cluster split analysis highlights the challenges of achieving strong performance in the context of a highly imbalanced dataset, where only 15% of students represent the minority class (STEM completers). In this context, metrics like F1-score and sensitivity are critical. The F1-score provides a balanced measure of precision (the proportion of predicted STEM completers that are true completers) and recall (the proportion of actual completers correctly identified), making it particularly valuable for evaluating models on imbalanced data. Sensitivity is especially important for ensuring that the model correctly identifies students who successfully completed a STEM degree. While some models performed well on specific metrics, the overall results suggest that significant improvements are needed to reliably predict outcomes for underrepresented groups (Figure 5.4).

Neural Nets achieved the highest F1 score of 0.65 and the highest specificity (0.91) among all models. This indicates Neural Nets' ability to correctly classify non-completers while maintaining a sensitivity of 0.71. In an educational context, this is particularly relevant for intervention programs, as high specificity ensures fewer false positives (non-completers mistakenly classified

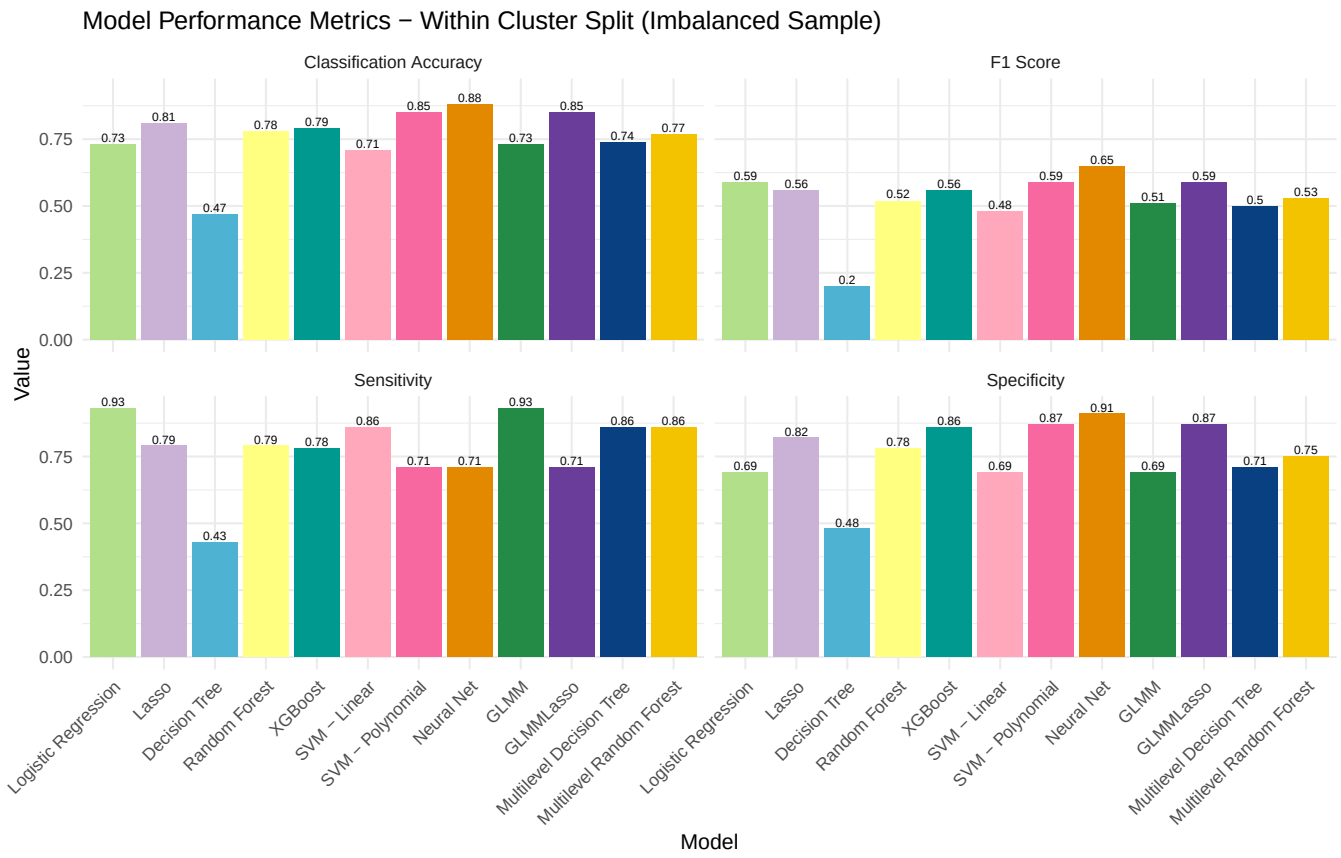

Figure 5.4: Performance metrics for standard and multilevel models, evaluated on their ability to predict STEM persistence using an imbalanced sample with an 85:15 class distribution with a within-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

as completers), optimizing resource allocation and targeting the right students. However, Logistic Regression and GLMM achieved the highest sensitivity (0.93), demonstrating their strong capacity to identify STEM completers, the minority class. This came at the expense of specificity (0.69), indicating a tendency to misclassify non-completers as completers. Other models also showed moderate success with sensitivity. SVM with a linear kernel, multilevel decision trees, and multilevel random forests each achieved sensitivity of 0.86, though their specificity and F1 scores were not as strong. Consistent with the simulation study findings, standard decision trees

performed poorly, with outcome metrics ranging from as low as 0.20 to only 0.48, further confirming their unsuitability for imbalanced hierarchical datasets. The inclusion of cluster IDs as fixed effects improved model performance for some metrics (Figure 5.5). XGBoost demonstrated the most significant gains, with sensitivity increasing from 0.78 to 0.86 and classification accuracy improving to 0.86, while maintaining specificity at 0.86. Random Forests also saw a notable increase in sensitivity, from 0.79 to 0.86, although their specificity dropped slightly from 0.78 to 0.75. Logistic Regression experienced a decrease in sensitivity to 0.71, but its specificity improved to 0.82, resulting in a more balanced performance. These results highlight the potential benefits of leveraging cluster-level information to capture hierarchical structure and improve predictions in imbalanced datasets. However, even with these improvements, the overall performance of the models remained suboptimal, emphasizing the need for innovative strategies, such as alternative modeling approaches or the inclusion of richer predictors, to better address extreme class imbalance in educational settings.

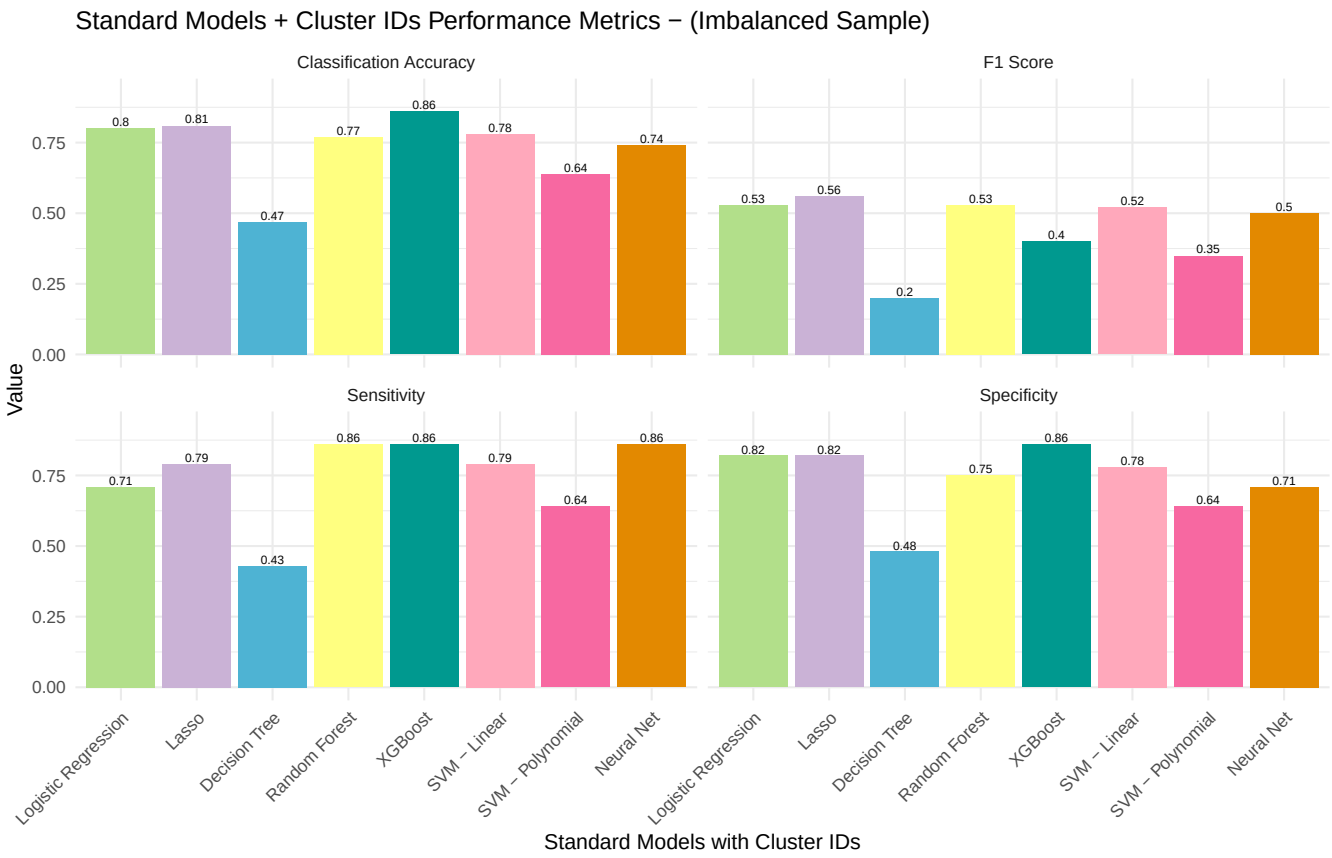

Figure 5.5: Performance metrics for standard models with cluster IDs as fixed effects, evaluated on their ability to predict STEM persistence using an imbalanced sample with an 85:15 class distribution and a within-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

5.3.2 Train-Test Split Between Cluster

The between-cluster split condition presented additional challenges, with model performance declining across most metrics due to the absence of shared cluster representation in the training and testing sets. These declines were particularly evident in sensitivity, which suffered for most models, reflecting the difficulty of identifying the minority class (STEM completers) in this more complex scenario (Figure 5.6).

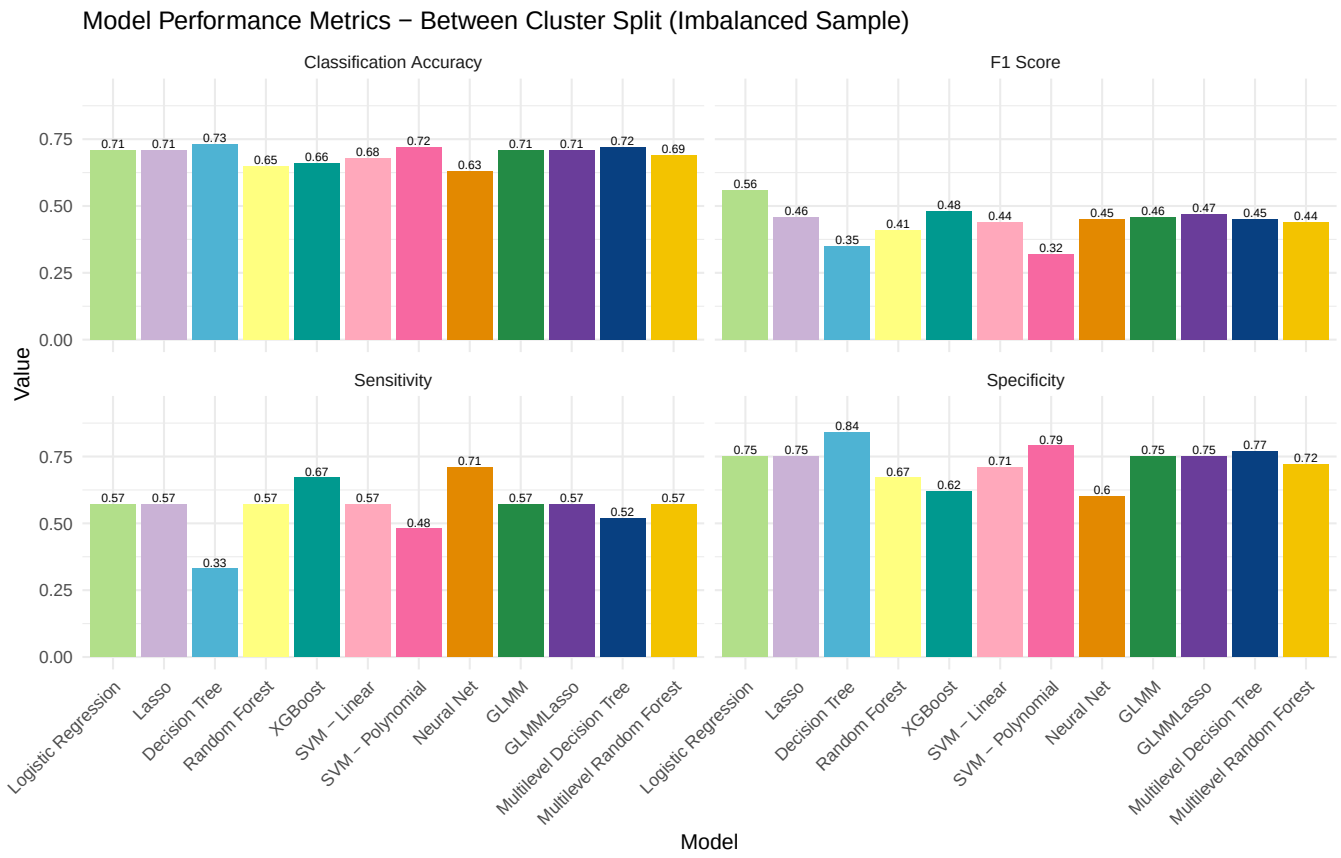

Figure 5.6: Performance metrics for standard and multilevel models, evaluated on their ability to predict STEM persistence using an imbalanced sample with an 85:15 class distribution with a between-cluster train-test split. Metrics displayed include classification accuracy, F1 score, sensitivity, and specificity.

Neural Nets demonstrated the best overall performance in identifying the smaller class, achieving the highest sensitivity (0.71) among all models. However, its specificity (0.60) revealed limitations in correctly classifying non-completers which was reflected in its F1 score (0.45). XGBoost followed closely, with a sensitivity of 0.67 and the second highest F1 score (0.48), but it too struggled with specificity (0.62). Other models, including logistic regression, Lasso, GLMM, and GLMMLasso, displayed average specificity (0.75) and classification accuracy (0.71). However, their sensitivity scores only reached 0.57, underscoring their limited utility

in predicting the smaller class. Standard decision trees performed slightly better with respect to specificity (0.84); however, their sensitivity scores (0.33) revealed significant challenges in identifying STEM degree completers, making them unsuitable for applications where predicting the smaller class is critical. These findings highlight the important considerations required when working with imbalanced datasets and unseen clusters. While some models, such as Neural Nets and XGBoost, show promise in addressing these challenges, the results underscore the need for careful predictive modeling approaches to achieve consistent performance across both classes.

The findings of this analysis illustrate the challenges of addressing extreme class imbalance in hierarchical educational datasets, where the minority class represents a small but critical subset of students. While Logistic Regression and Neural Nets demonstrated strengths in sensitivity, and XGBoost showed relatively balanced performance across sensitivity and specificity, no single model emerged as consistently strong across all metrics. This underscores the inherent complexity of working with imbalanced data. These trade-offs between sensitivity and specificity must be carefully evaluated based on the specific objectives of the intervention or policy initiative. Additionally, the persistent challenges observed in the between-cluster split condition indicate that traditional modeling approaches may not be sufficient for generalizing to unseen clusters. This limitation has important implications for applications where models are expected to predict outcomes for students in new schools or programs. Finally, the generally moderate to poor performance across models highlights the need for ongoing research and methodological innovation. Predictive models must not only address class imbalance but also account for the unique challenges posed by hierarchical data structures. Future research should explore the integration of equity-focused metrics, such as fairness and bias assessments, to ensure that models are not only accurate but also equitable in their predictions for underrepresented groups.

5.4 Summary

The analyses in this chapter provided important insights into the context-specific strengths and limitations of standard and multilevel machine learning models for predicting STEM degree completion. Model performance varied significantly depending on the dataset characteristics and evaluation context. In the full-sample within-cluster analysis, Logistic Regression and both SVM models demonstrated strong sensitivity, excelling at correctly identifying STEM degree completers. Meanwhile, XGBoost stood out for its high specificity, making it particularly effective for identifying non-completers. In contrast, the full-sample between-cluster analysis highlighted GLMM's ability to predict STEM degree noncompleters. Similar to the within-cluster analysis, both SVM models demonstrated their ability to indentify students likely to complete their STEM degree. The imbalanced data analyses revealed different patterns, with Neural Nets and XGBoost often performing well relative to other models. These findings underscore the difficulty of addressing class imbalance and the necessity of carefully selecting models based on the specific goals of the analysis. Across all contexts, no single model consistently excelled across all metrics, reinforcing the importance of tailoring predictive approaches to the unique characteristics of each dataset.

These empirical analyses also mirrored findings from the simulation study, demonstrating that incorporating cluster IDs as fixed effects can improve standard model performance, highlighting the importance of contextual information in predictive modeling and the potential effectiveness of standard models in hierarchical settings.

Chapter 6: Discussion

6.1 Overview of Findings

This dissertation explored the performance of standard and multilevel machine learning algorithms in hierarchical data scenarios, emphasizing predictive modeling. Through simulation studies and empirical analyses, the results revealed that while multilevel models generally excel under conditions with high clustering effects (e.g., high residual level-2 variance), their advantages diminished when these effects were less pronounced, aligning with prior studies that suggested their utility is contingent on the strength of cluster-level dependencies. Standard algorithms, especially those that incorporated fixed effects for cluster IDs, frequently closed the performance gap, offering practical and computationally efficient alternatives in many scenarios.

The empirical analyses further highlighted the challenges of addressing class imbalance, especially in predicting STEM persistence. Even with optimized thresholds, most models demonstrated limited sensitivity and F1 scores for the minority class. Logistic Regression and GLMM stood out for their strong sensitivity in identifying STEM completers, albeit at the cost of specificity, while Neural Nets and XGBoost offered more balanced but moderate performance across metrics. These findings underscore the complexity of predictive tasks in hierarchical and imbalanced datasets.

6.2 Contributions to the Literature

The findings of this dissertation contribute to the existing body of research on predictive modeling in hierarchical data in several ways. First, by evaluating advanced standard algorithms alongside multilevel methods, this study provided a broader perspective on the range of tools available for hierarchical data analysis. While prior studies have primarily focused on traditional algorithms like logistic regression or random forests, this research also considered more advanced standard algorithms such as XGBoost, SVM, and Neural Nets, which had not been extensively explored in hierarchical data contexts. This work also highlighted the potential of alternative approached to handle the complexities of hierarchical data effectively. In particular, the results demonstrated that including cluster IDs as fixed effects in standard models enhances their performance, bridging the gap with multilevel models even in datasets with high clustering effects.

Second, this dissertation addressed gaps in the literature by exploring model performance under a wide range of data-generating conditions, including variations in residual level-2 variance, the complexity of the data generating model, the number of clusters and individuals per cluster, and training/testing split mechanisms. These analyses offer practical guidance for researchers, particularly in fields like education, where hierarchical data structures are prevalent. Finally, by highlighting the limitations of both multilevel and standard methods in generalizing to unseen clusters, this work provides a critical perspective on the challenges of hierarchical prediction, even with appropriate multilevel modeling techniques.

6.3 Practical Implications for Educational Researchers

The findings from this dissertation provide several practical insights into the application of machine learning for educational researchers and policymakers working with hierarchical data. In contexts where outcomes are strongly influenced by cluster membership, such as datasets with high ICC values or substantial residual level-2 variance, multilevel models like GLMM remain essential tools for capturing hierarchical dependencies and ensuring accurate and stable predictions. However, this research also demonstrated that standard models incorporating fixed effects for cluster IDs, such as XGBoost, Neural Nets, and simpler methods like Logistic Regression, often perform comparably to multilevel models. These results also extend recommendations suggesting that cluster IDs as fixed effects are appropriate for datasets with a small number of clusters, showing instead that this approach remains viable even in datasets with large numbers of clusters. Consequently, researchers can prioritize computationally efficient standard models when interpretability and ease of implementation are important considerations.

Addressing the challenges posed by extreme class imbalance requires careful model selection and evaluation. The empirical analyses showed that Logistic Regression, with its high sensitivity, is effective for identifying underrepresented groups, but this often comes at the expense of specificity. Conversely, models like Neural Nets and XGBoost offer more balanced performance across sensitivity and specificity, making them valuable for contexts where identifying at-risk students while minimizing false positives is critical. These findings emphasize the importance of selecting models that align with specific research or policy goals, such as maximizing inclusion in intervention programs or targeting resources efficiently.

The persistent challenges observed in the between-cluster split condition highlight the dif-

ficulties of generalizing predictions to unseen clusters. Incorporating additional level-2 predictors, such as institutional characteristics or program-level data, may help mitigate these issues by reducing unexplained variability at the cluster level. To aid researchers and policymakers in making informed decisions about model selection, a flowchart summarizing recommendations for predictive tasks is provided in Figure 6.1. This decision tool outlines key considerations, such as whether predictions are being made for new clusters and the availability of high-quality level-2 predictors, and highlights model options best suited to different scenarios.

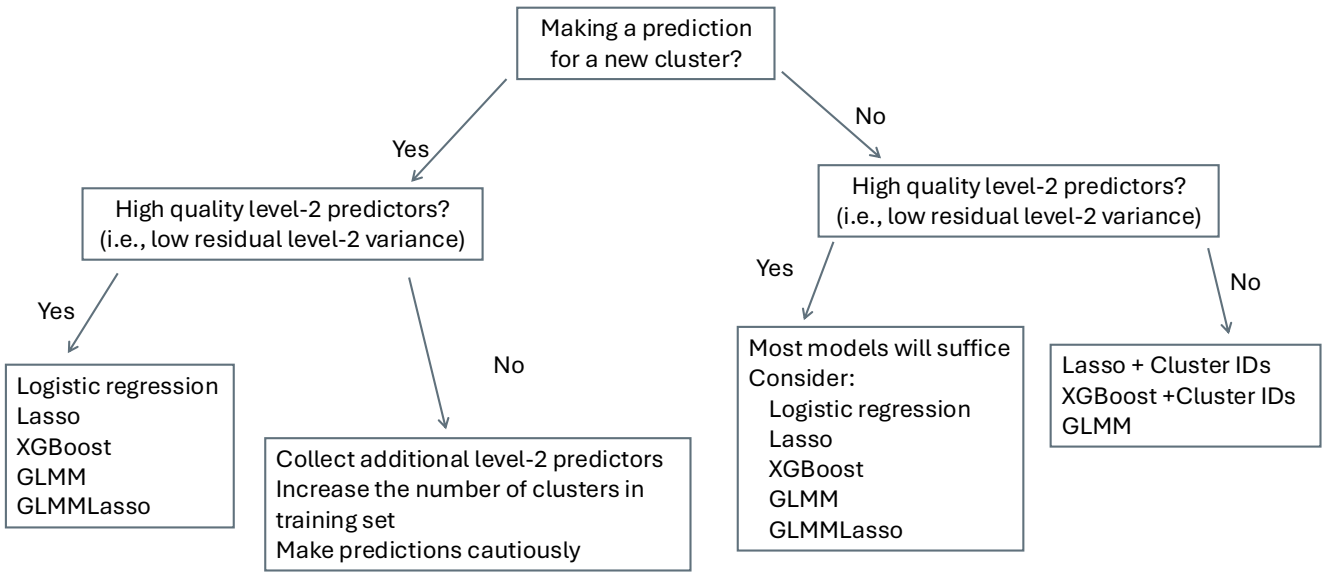

Figure 6.1: Decision-making flowchart for selecting predictive models for hierarchical data.

6.4 Limitations and Future Directions

While this research provides valuable insights into predictive modeling in hierarchical data, it is important to acknowledge several limitations that could inform future studies. The empirical analyses relied on two educational datasets with relatively modest ICC values ranging from 0.10 to 0.11. Although these datasets reflect realistic conditions in some educational contexts, the findings may not generalize to settings with stronger clustering effects. Future research should explore model performance across datasets with a broader range of ICC values to better understand the interaction between clustering effects and algorithm effectiveness. Additionally, the limited number of clusters in the datasets may have influenced the observed model performances, highlighting the need for future studies to evaluate empirical datasets with larger numbers of clusters to assess how this factor impacts predictive accuracy.

Another limitation lies in the study's focus on classification tasks, leaving regression analyses unexplored. Extending this research to include regression tasks could provide a more comprehensive understanding of how standard and multilevel methods perform across different predictive contexts. Similarly, while this dissertation emphasized overall predictive performance, it did not evaluate the interpretability or usability of the models in real-world decision-making. Future studies could incorporate feedback from practitioners and stakeholders to assess the practical implications of using these models for policy and intervention design.

Both the simulation study and empirical analyses highlighted the difficulty of generalizing predictions to unseen clusters, a significant limitation for hierarchical modeling. Developing innovative approaches to enhance generalization represents a promising direction for future research. Additionally, the simulation study did not vary the random slope (σ^2_{u1}) or provide certain

models with the correct functional form for nonlinear data-generating processes. Similarly, the study used a multilevel random intercept model but did not include random slope models. The absence of random slopes may have limited the ability of the multilevel models to fully capture variability in the relationships between predictors and outcomes across clusters if the true data-generating model included random slopes. Addressing these limitations could yield more nuanced insights into model performance and applicability.

Future research could also evaluate the performance of large language models (LLMs) in hierarchical contexts. Given their increasing prominence in predictive modeling, LLMs may also perform well on hierarchical datasets. Finally, future work should explore equity-focused metrics, such as fairness and bias assessments, to ensure that predictive models not only achieve strong performance but also promote equitable outcomes across diverse populations.

Appendix A: Appendix

Table A.1: Classification Accuracy ANOVA Results by Training-Testing Split Condition

Factor	Degrees of Freedom	F-Statistic	P-Value	η^2
Training-Testing Split				
Overall	1	79844.0	<0.001	0.058
Within-Cluster Split				
Model	11	317371.0	<0.001	0.844
DGP	5	69.2	<0.001	0.001
Residual Level-2 Variance	2	3288.0	<0.001	0.010
J	2	117.1	<0.001	<0.001
n_j	1	1526.0	<0.001	0.002
DGP*Model	55	16.1	<0.001	<0.001
DGP*Residual Level-2 Variance	10	0.6	0.775	<0.001
DGP*J	10	0.8	0.664	<0.001
DGP*n_j	5	0.7	0.645	<0.001
Residual Level-2 Variance*Model	22	7670.0	<0.001	0.030
Residual Level-2 Variance*J	4	201.1	<0.001	0.001
Residual Level-2 Variance*n_j	2	60.9	<0.001	<0.001
J*Model	22	1378.0	<0.001	0.007
J*n_j	2	23.9	<0.001	<0.001
n_j*Model	11	841.3	<0.001	0.002
DGP*Residual Level-2 Variance*Model	110	2.0	<0.001	<0.001
DGP*Residual Level-2 Variance*J	20	0.2	1.000	<0.001
DGP*Residual Level-2 Variance*n_j	10	0.1	1.000	<0.001
Residual Level-2 Variance*J*Model	44	63.07	<0.001	<0.001
Residual Level-2 Variance*J*n_j	4	0.345	0.847	<0.001
Residual Level-2 Variance*n_j*Model	22	73.05	<0.001	<0.001
J*n_j*Model	22	45.01	<0.001	<0.001

Factor	Degrees of Freedom	F-Statistic	P-Value	η^2
Between-Cluster Split				
Model	11	91971.0	<0.001	0.611
DGP	5	50.85	<0.001	<0.001
Residual Level-2 Variance	2	51500.0	<0.001	0.138
J	2	2571.0	<0.001	0.008
n_j	1	39.8	<0.001	<0.001
DGP*Model	55	10.2	<0.001	<0.001
DGP*Residual Level-2 Variance	10	2.634	0.003	<0.001
DGP*J	10	1.193	0.290	<0.001
DGP*n_j	5	1.19	0.311	<0.001
Residual Level-2 Variance*Model	22	1772.0	<0.001	0.014
Residual Level-2 Variance*J	4	96.61	<0.001	0.001
Residual Level-2 Variance*n_j	2	38.84	<0.001	<0.001
J*Model	22	178.6	<0.001	0.002
J*n_j	2	19.31	<0.001	<0.001
n_j*Model	11	132.2	<0.001	0.001
Residual Level-2 Variance*J*Model	44	34.69	<0.001	0.001
Residual Level-2 Variance*J*n_j	4	4.278	0.002	<0.001
Residual Level-2 Variance*n_j*Model	22	30.68	<0.001	<0.001
J*n_j*Model	22	13.36	<0.001	<0.001

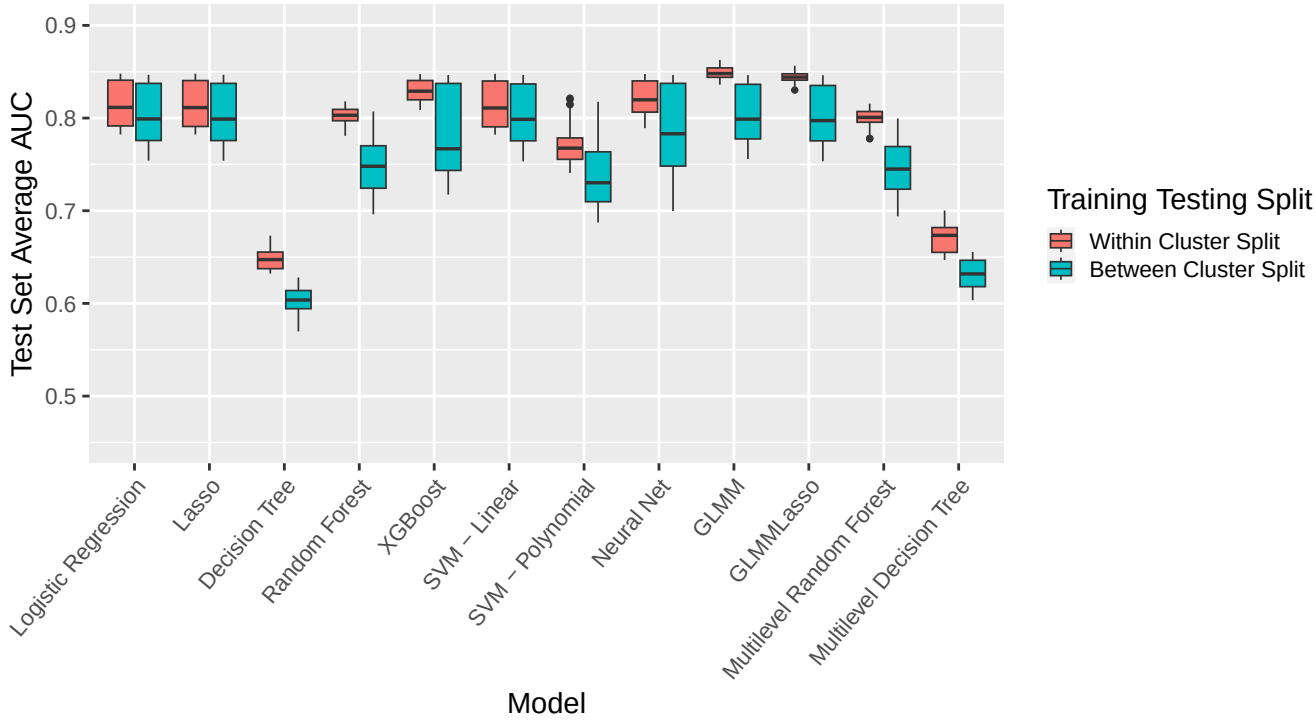

Figure A.1: Test set AUC by model and training-testing split.

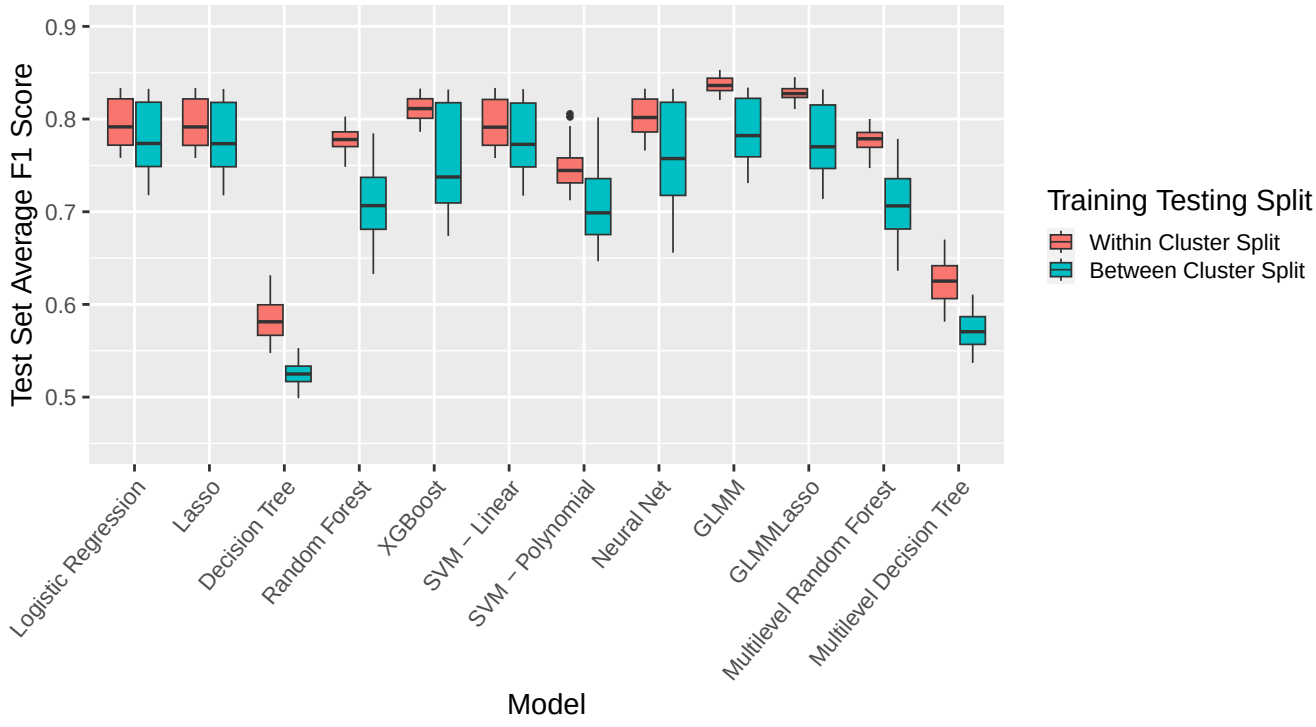

Figure A.2: Test set F1 Score by model and training-testing split.

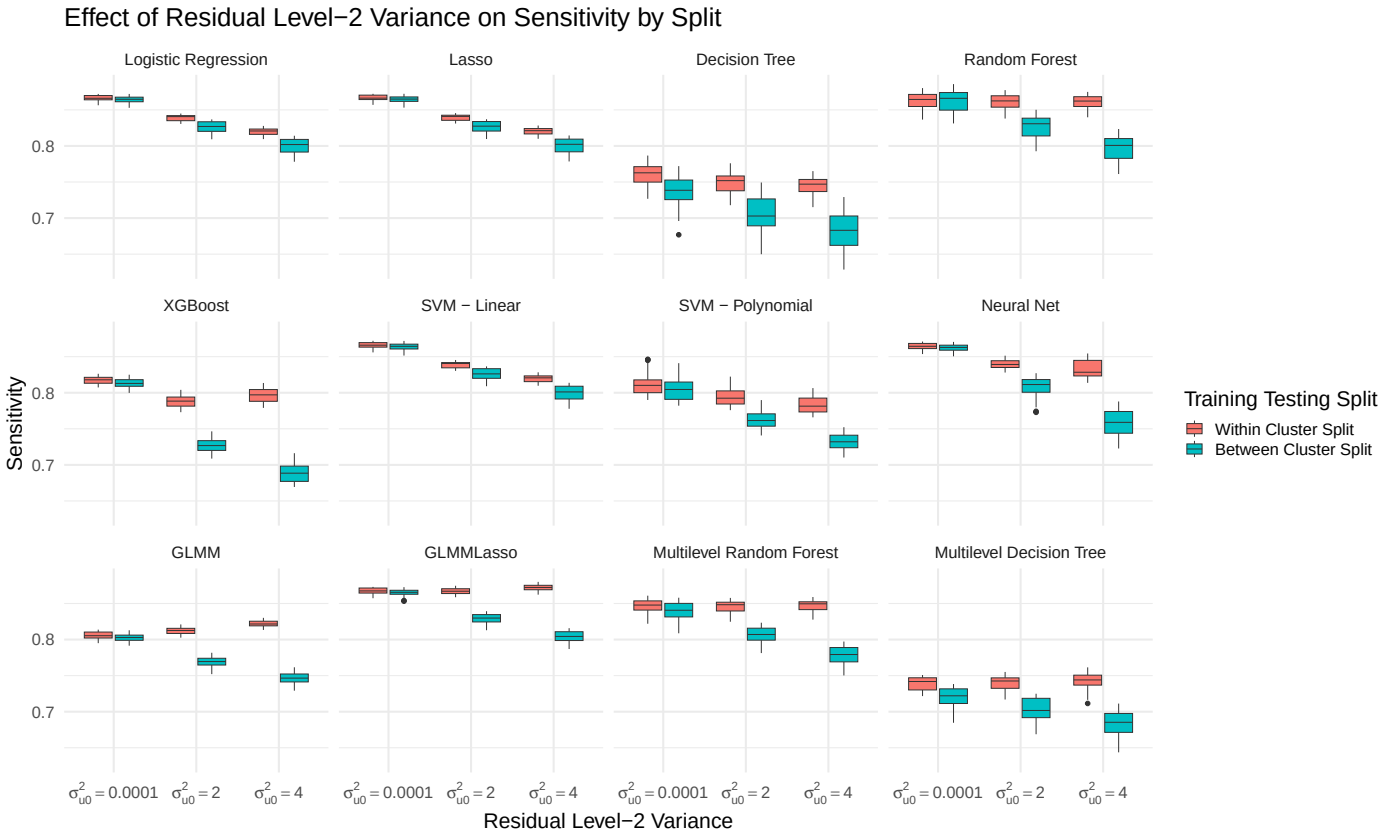

Figure A.3: Average test set sensitivity by model, training testing split, and across levels of residual level-2 variance.

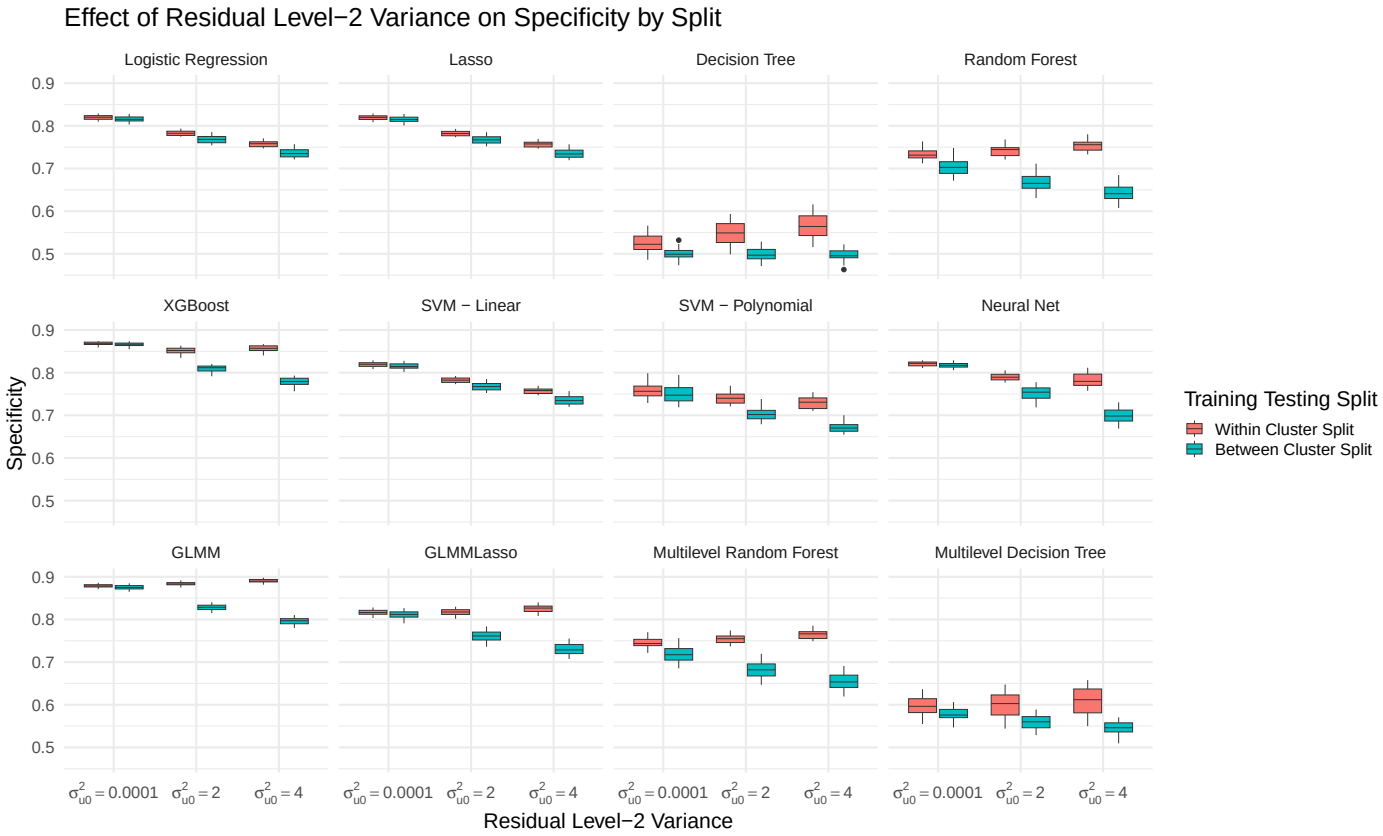

Figure A.4: Average test set specificity by model, training testing split, and across levels of residual level-2 variance.

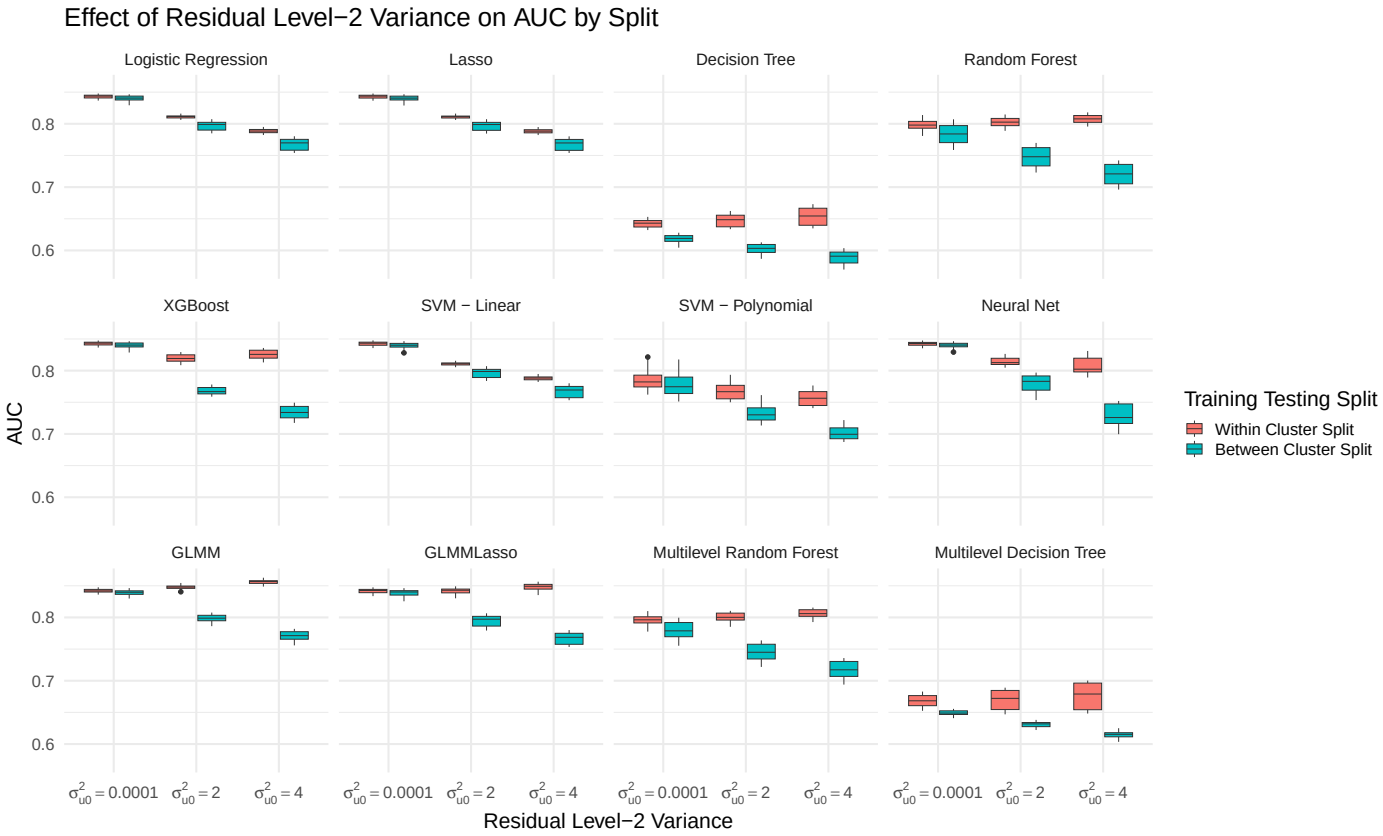

Figure A.5: Average test set AUC by model, training testing split, and across levels of residual level-2 variance.

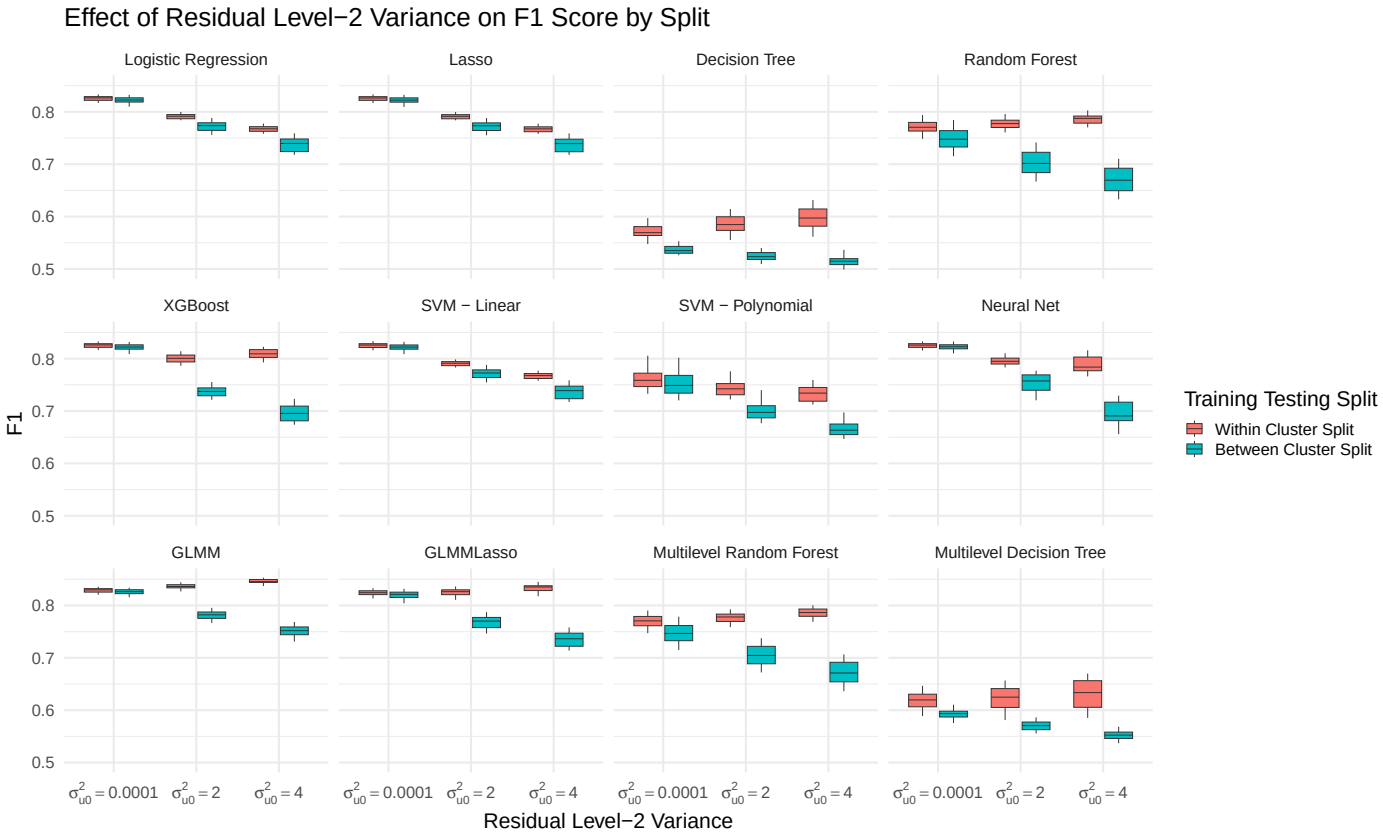

Figure A.6: Average test set F1 Score by model, training testing split, and across levels of residual level-2 variance.

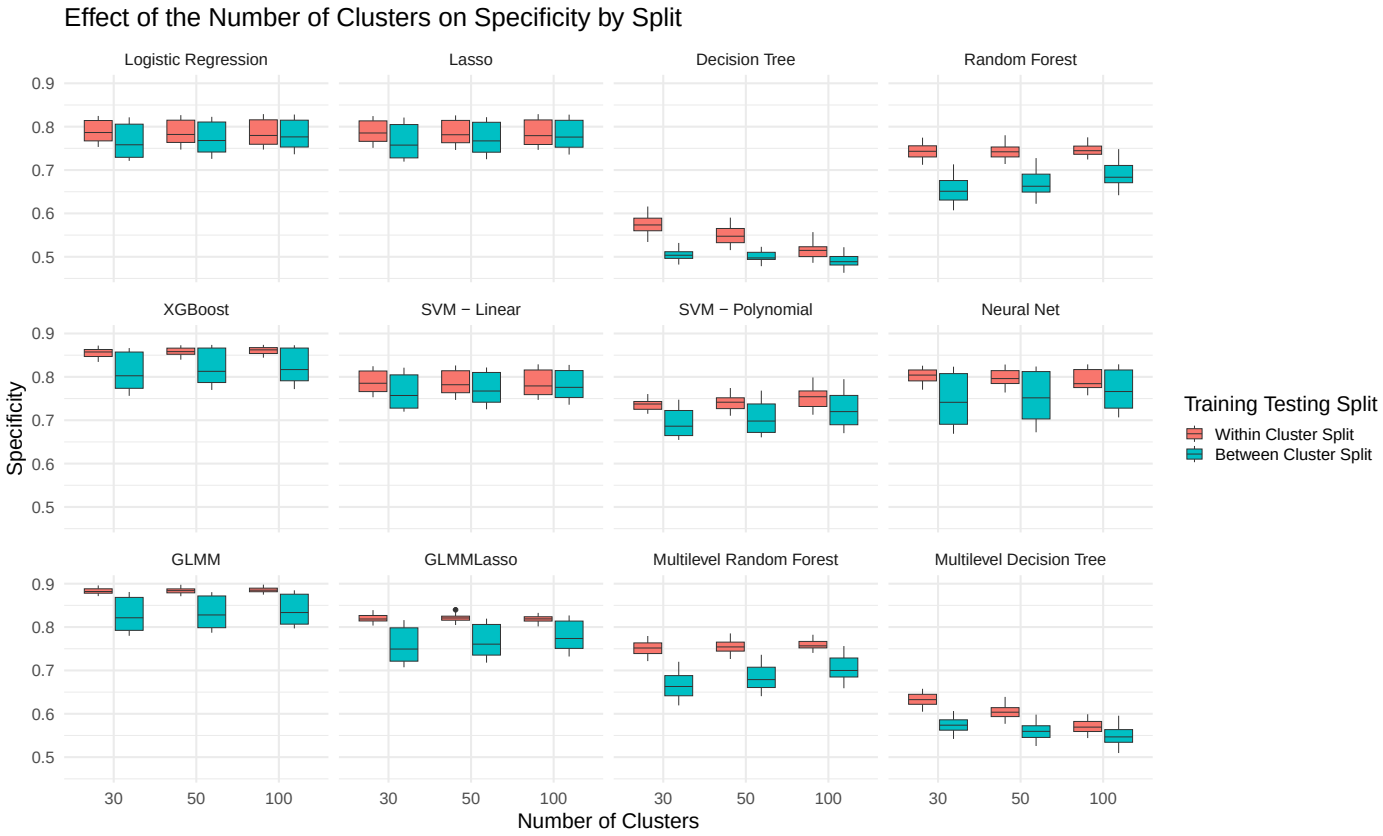

Figure A.7: Average test set specificity by model, training testing split, and across levels of number of groups.

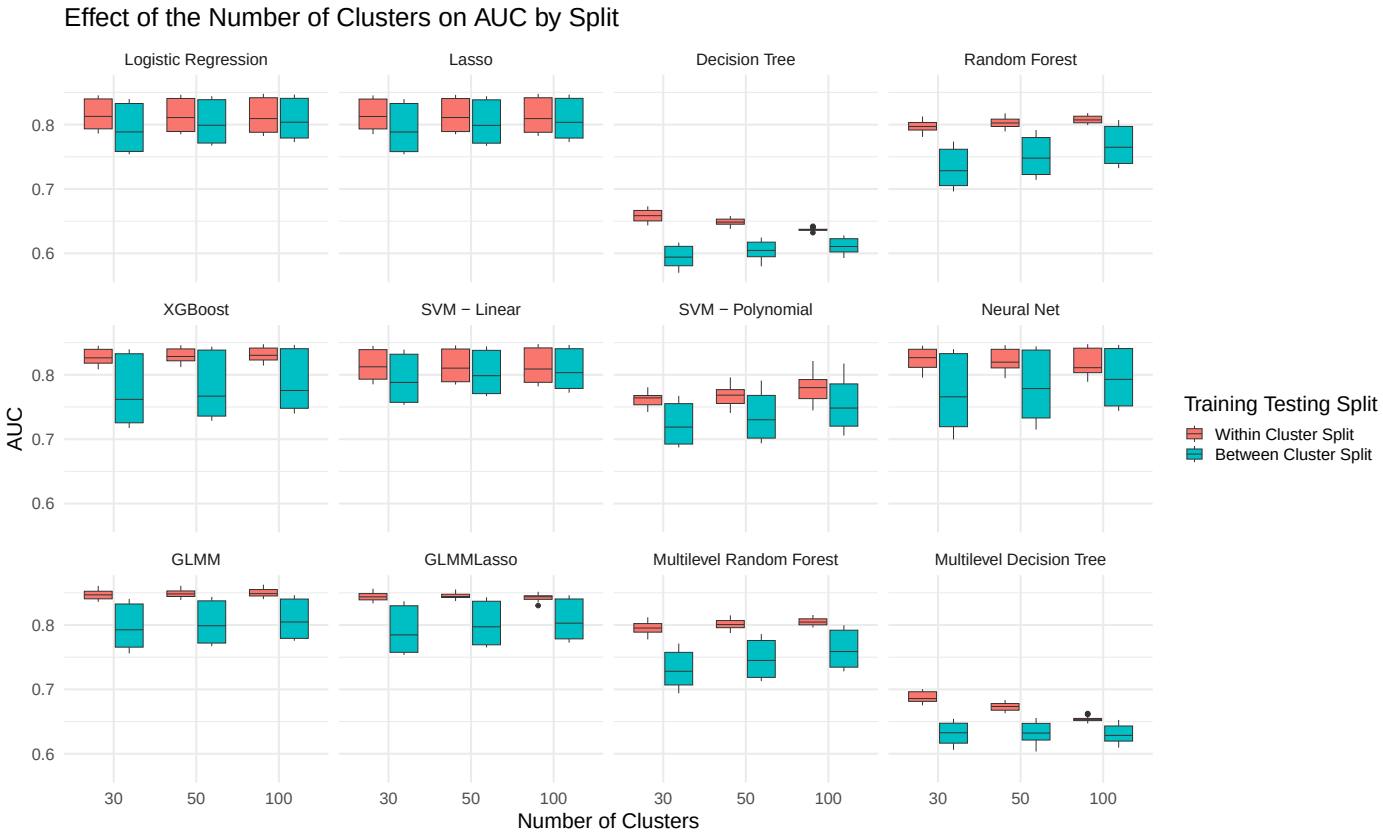

Figure A.8: Average test set AUC by model, training testing split, and across levels of number of groups.

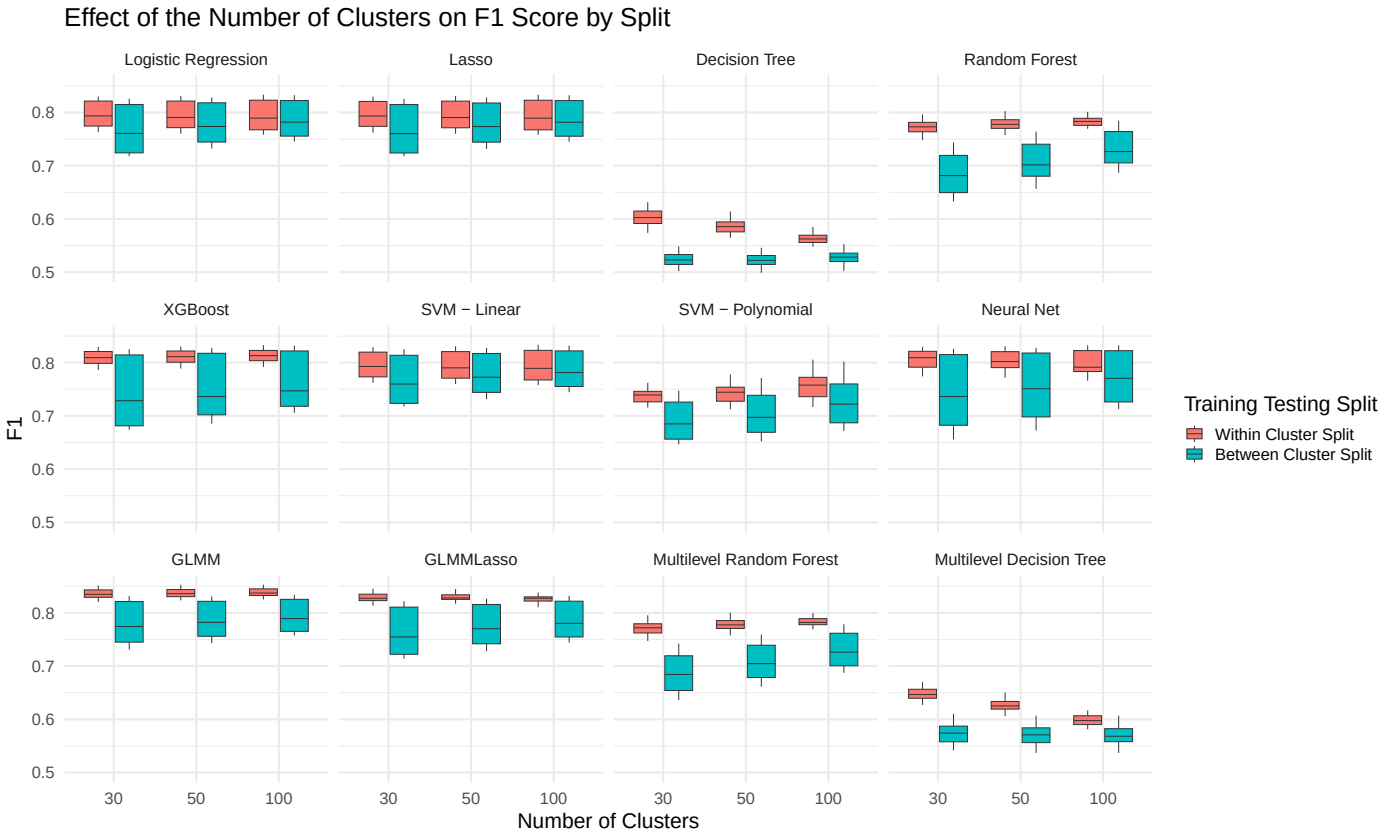

Figure A.9: Average test set F1 Score by model, training testing split, and across levels of number of groups.

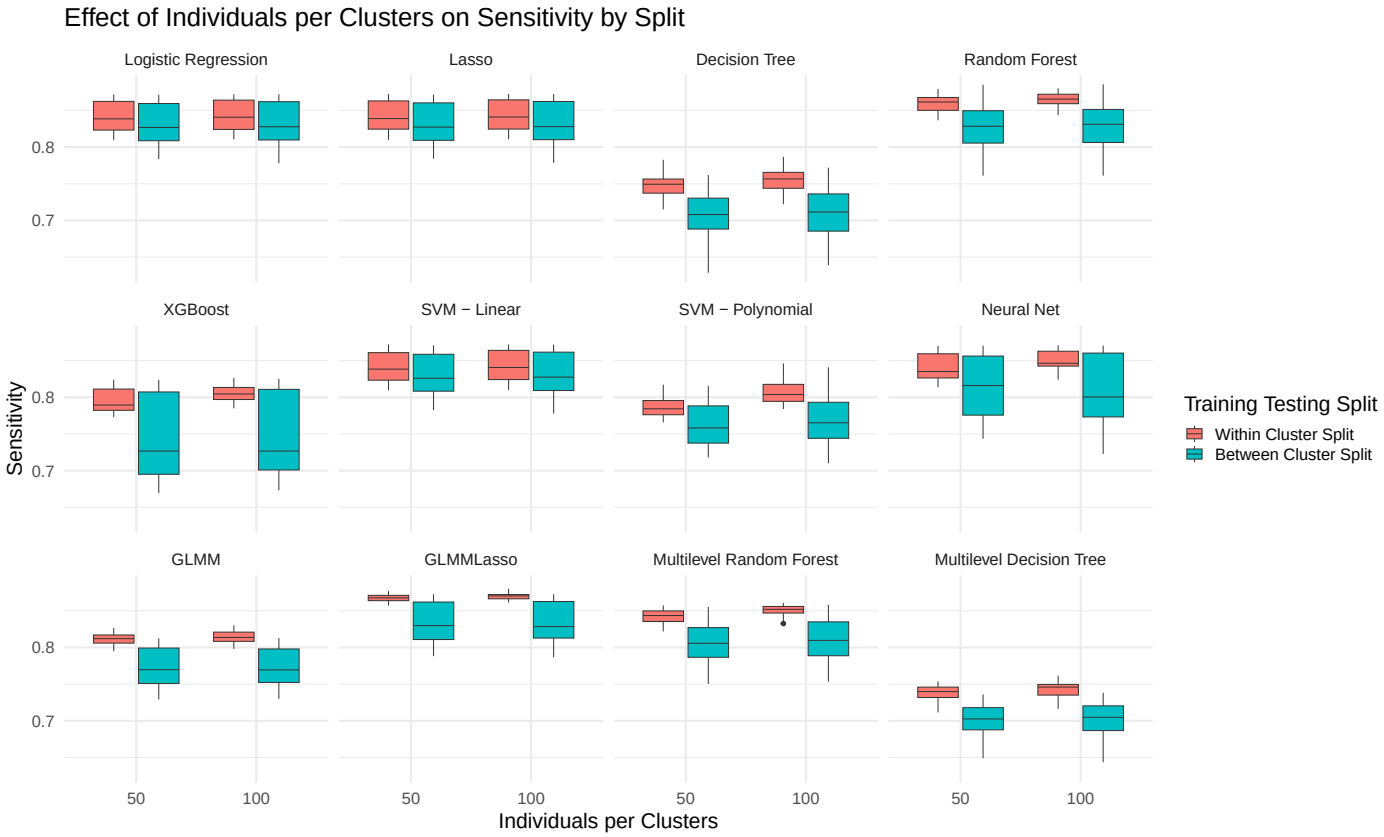

Figure A.10: Average test set sensitivity by model, training testing split, and across levels of individuals per group.

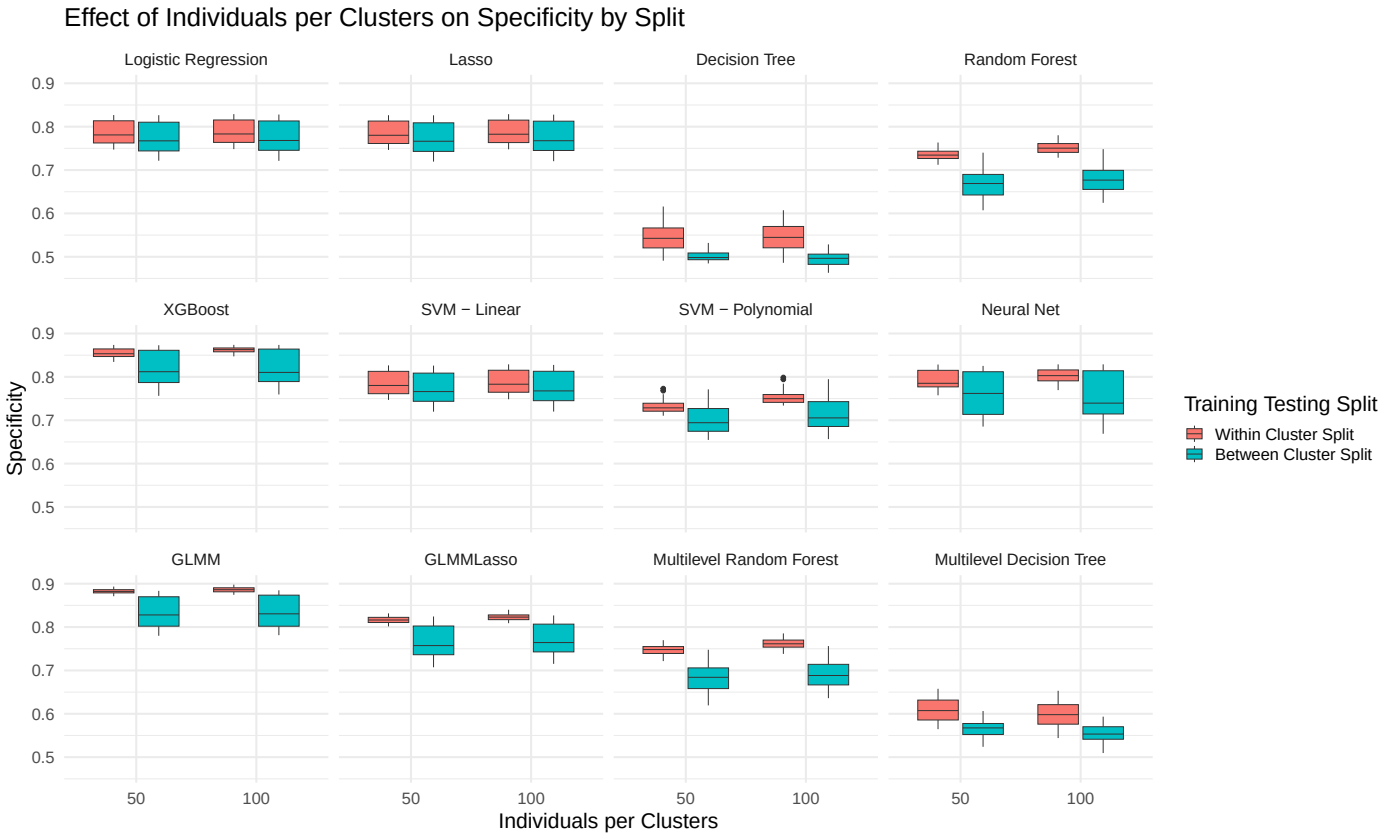

Figure A.11: Average test set specificity by model, training testing split, and across levels of individuals per group.

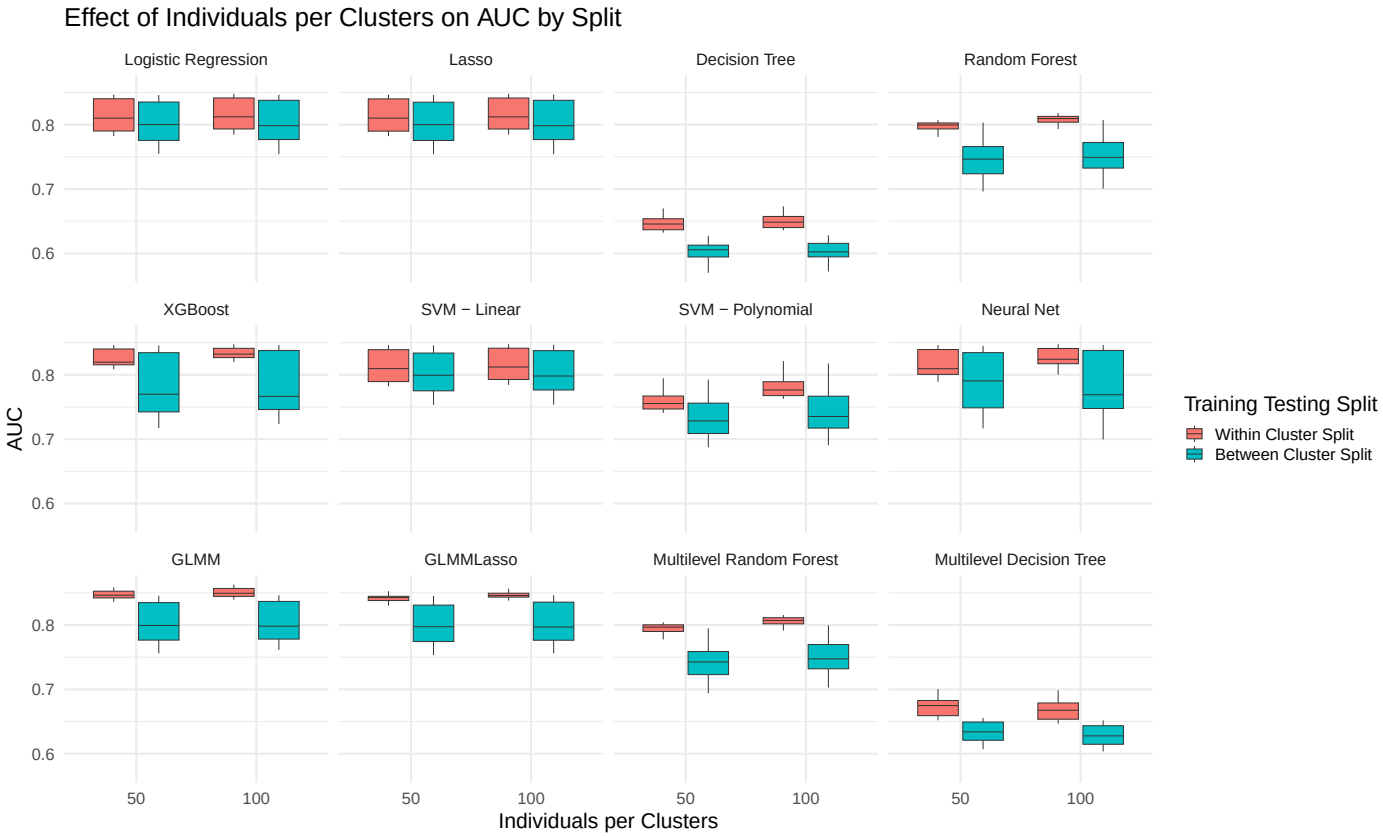

Figure A.12: Average test set AUC by model, training testing split, and across levels of individuals per group.

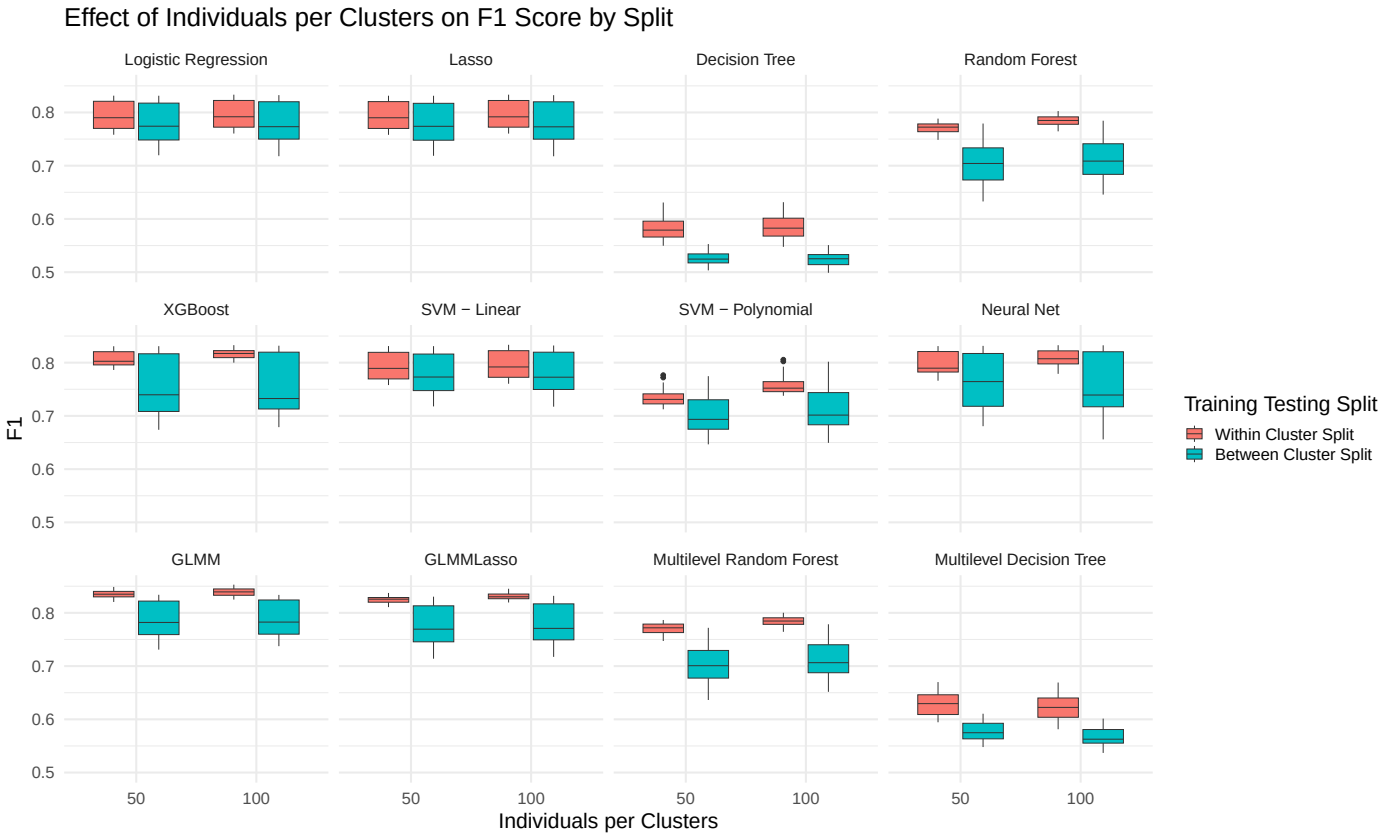

Figure A.13: Average test set F1 Score by model, training testing split, and across levels of individuals per group.

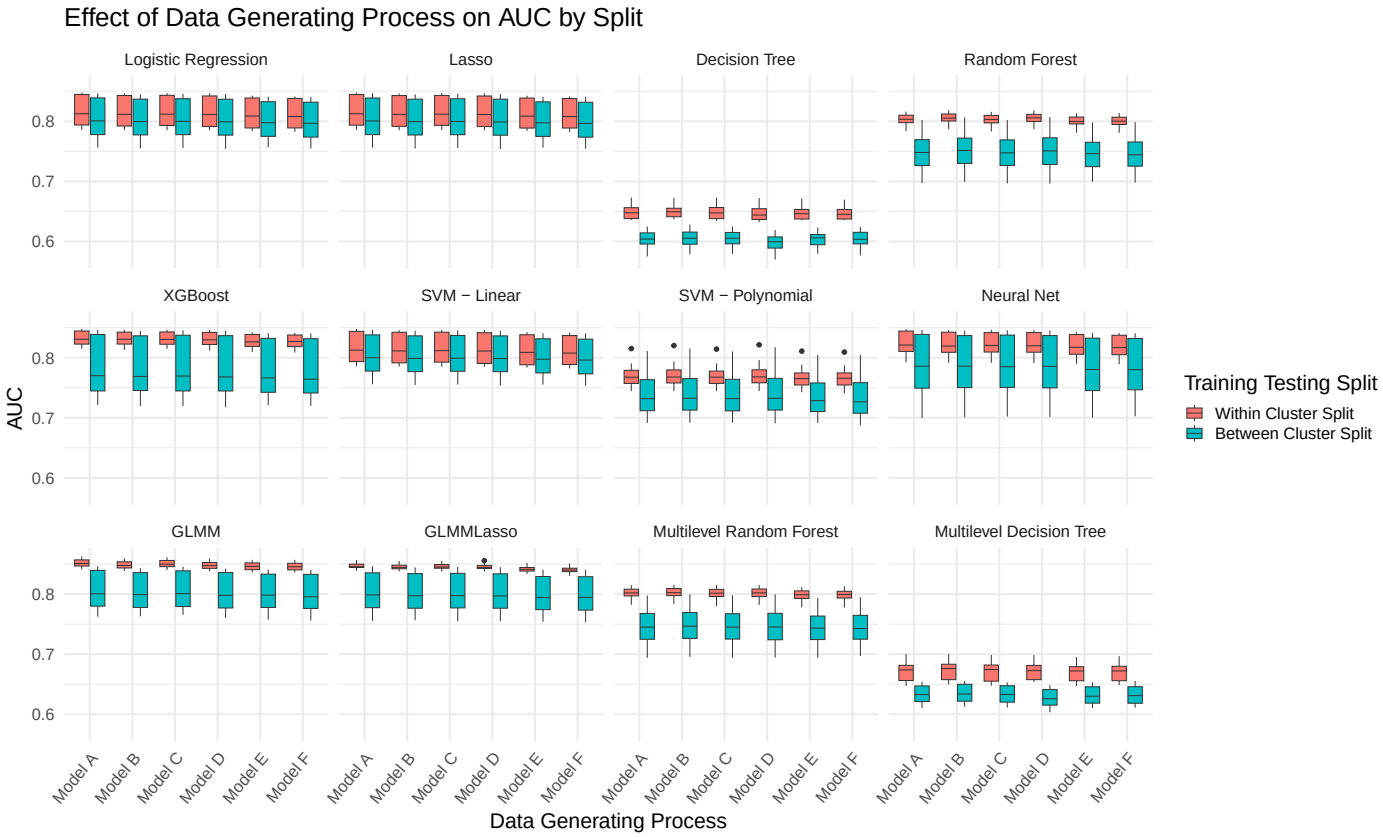

Figure A.14: Average test set AUC by model, training testing split, and across data generating processes.

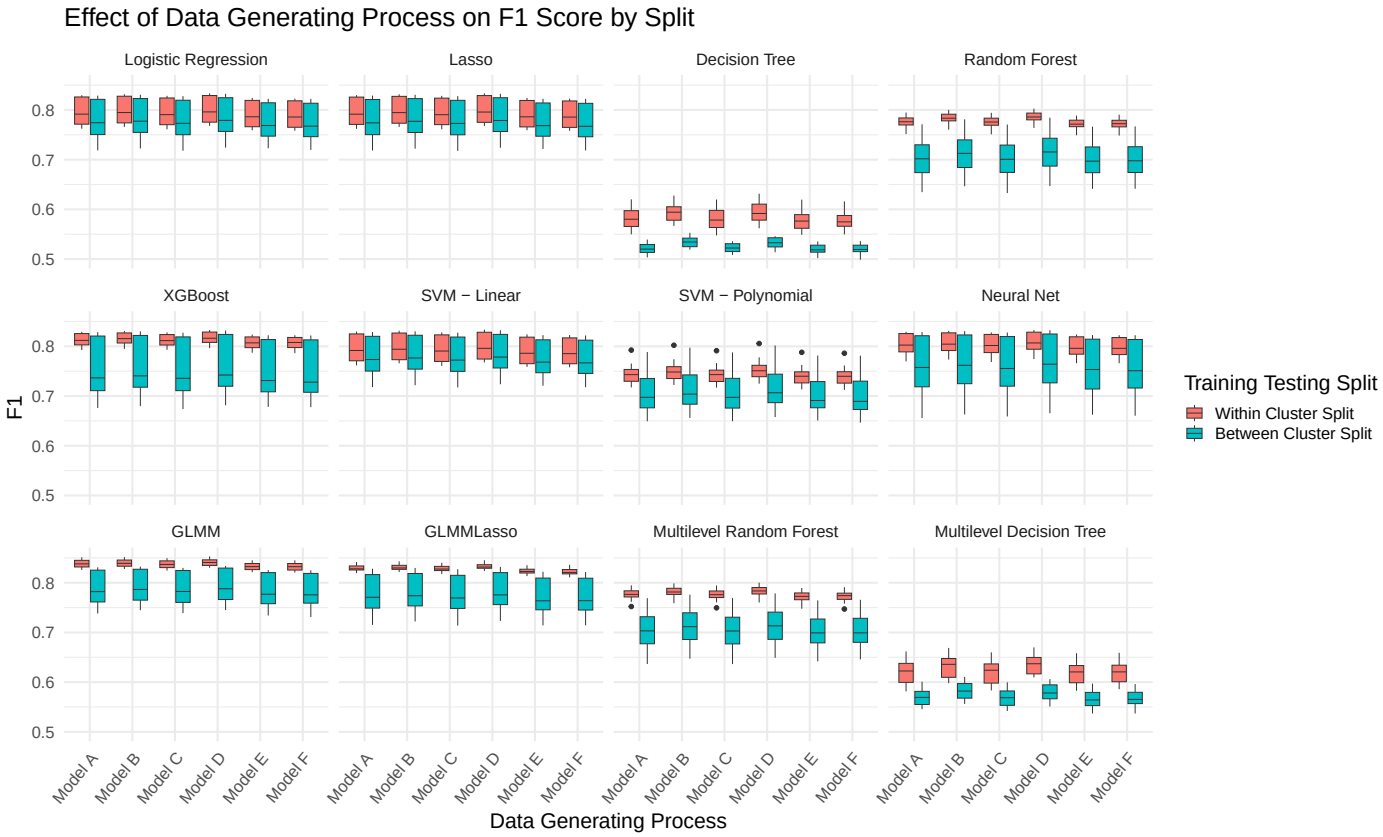

Figure A.15: Average test set F1 Score by model, training testing split, and across data generating processes.

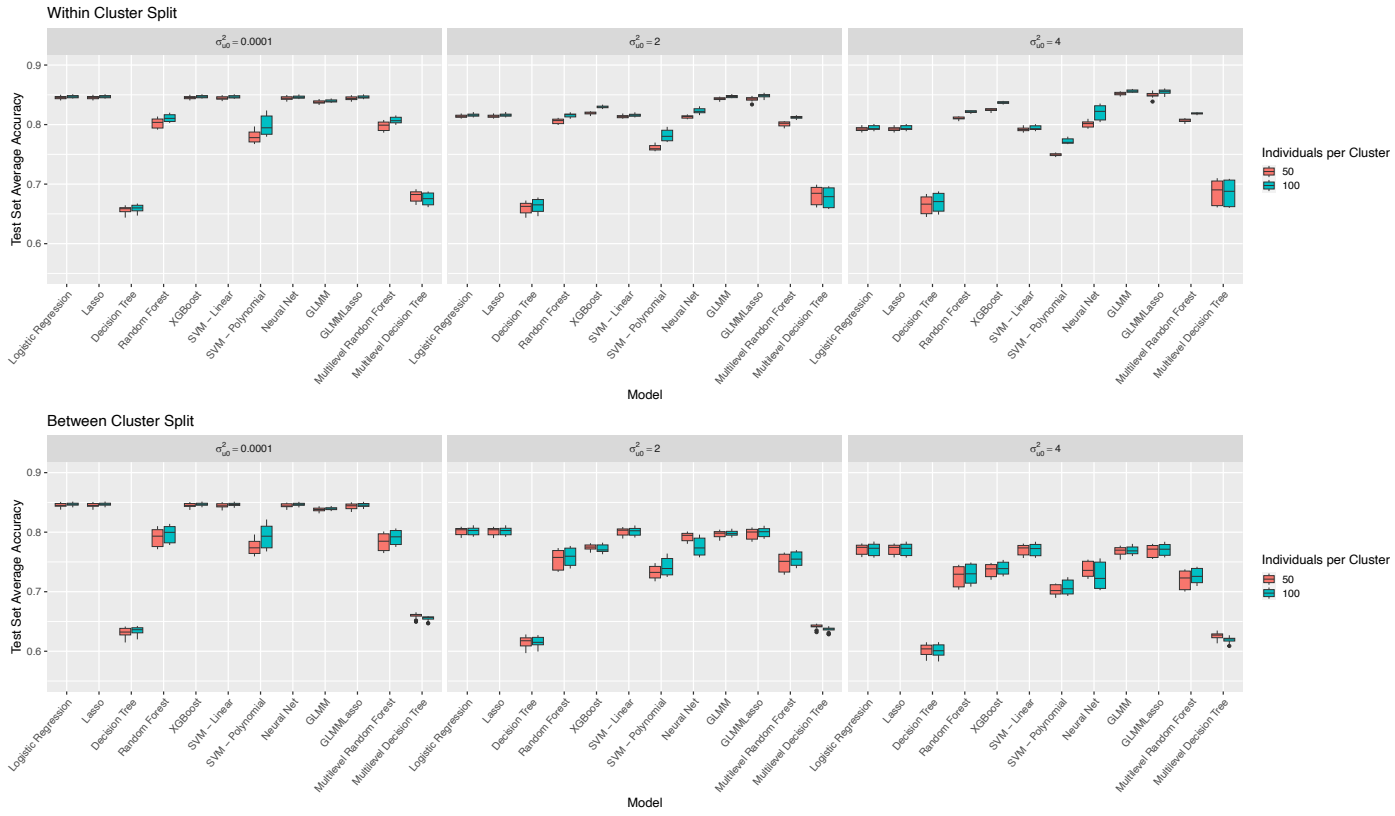

Figure A.16: Average test set accuracy by model, training testing split, residual level-2 variance and individuals per group.

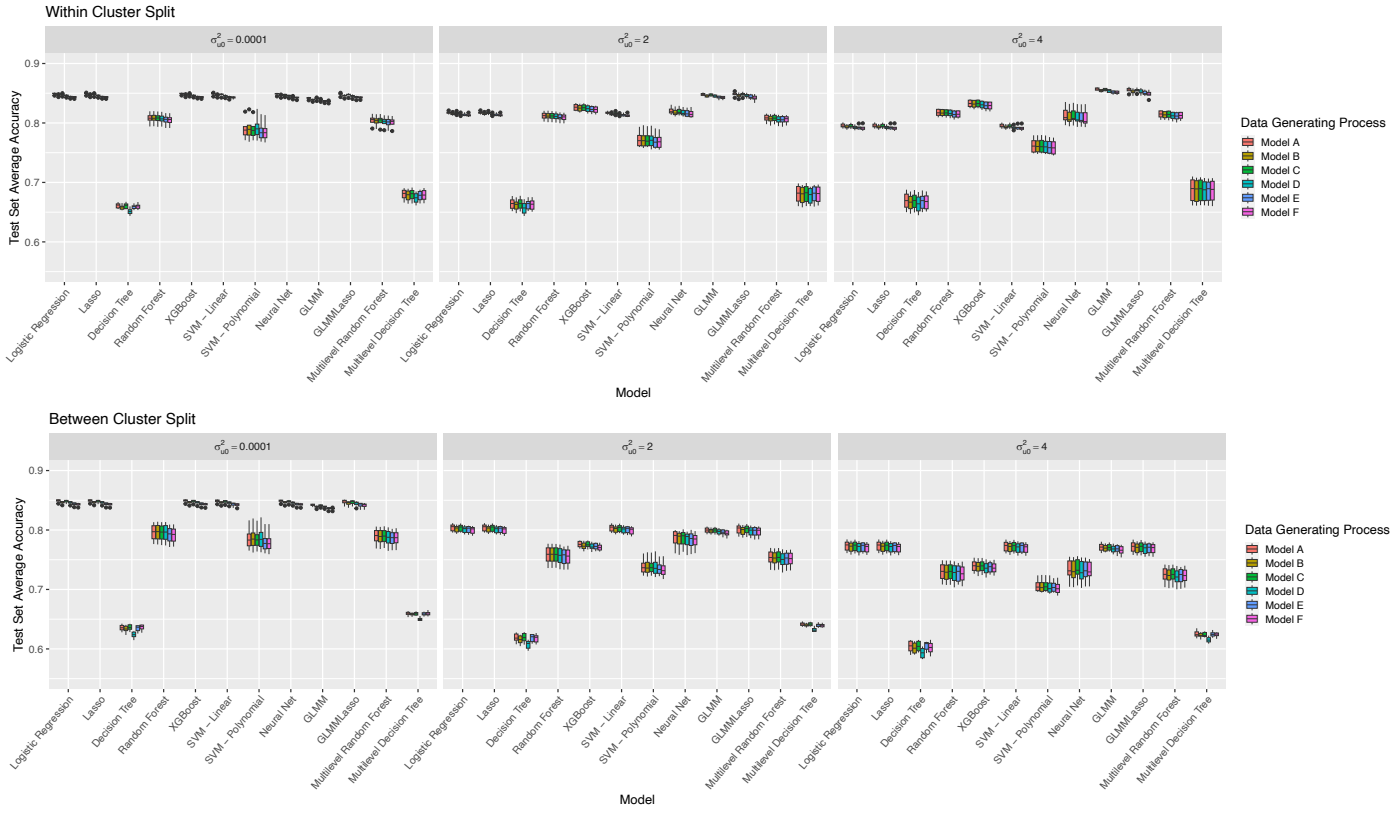

Figure A.17: Average test set accuracy by model, training testing split, residual level-2 variance and data generating process.

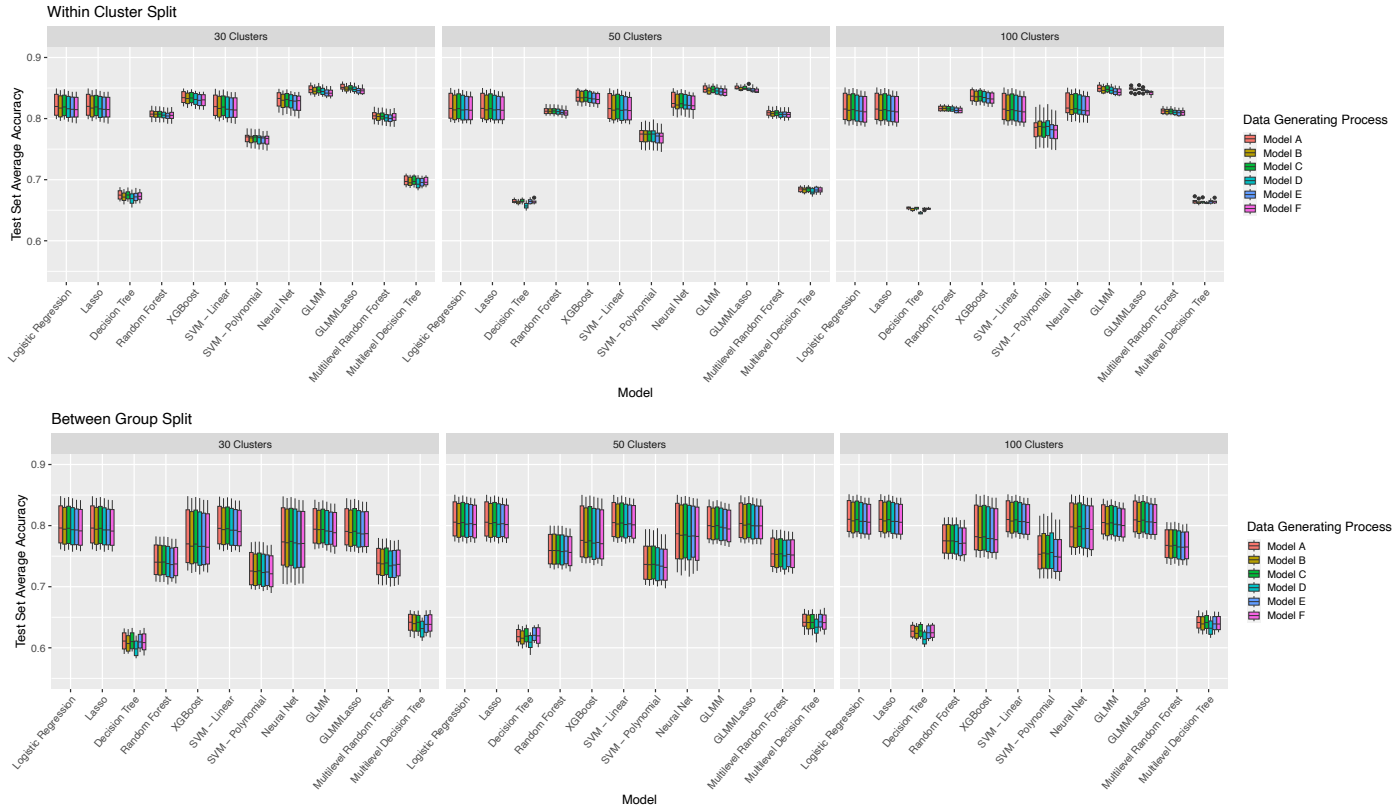

Figure A.18: Average test set accuracy by model, training testing split, number of groups and data generating process.

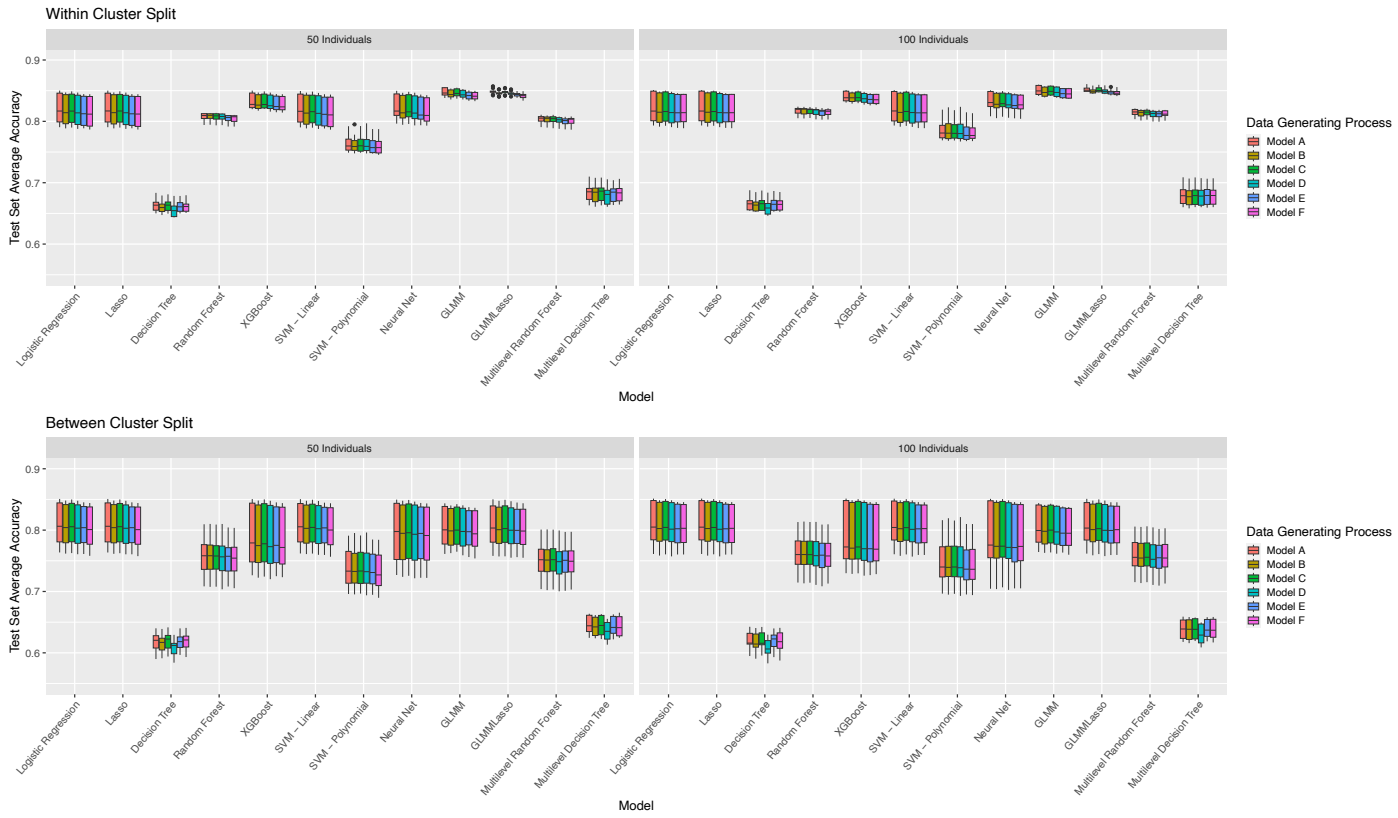

Figure A.19: Average test set accuracy by model, training testing split, individuals per group and data generating process.

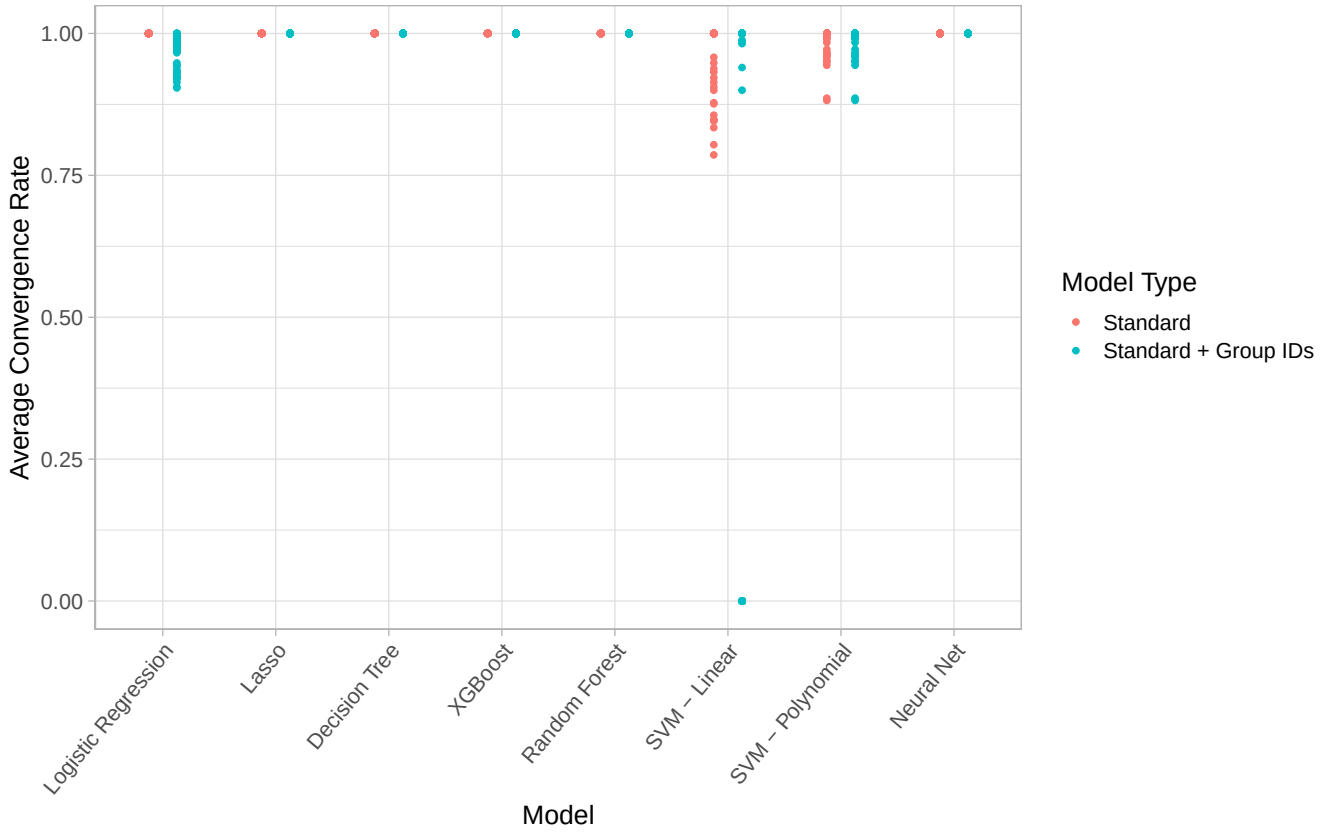

Figure A.20: Convergence rates of standard models with group IDs as fixed effects .

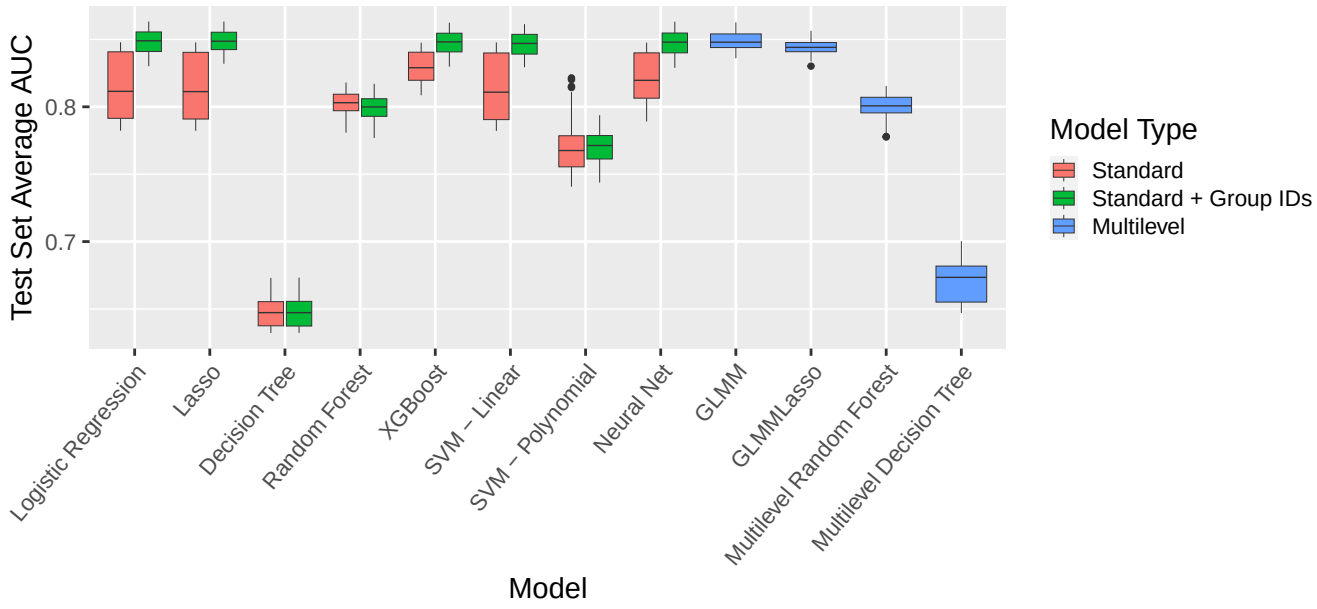

Figure A.21: Average test set AUC by model and model type.

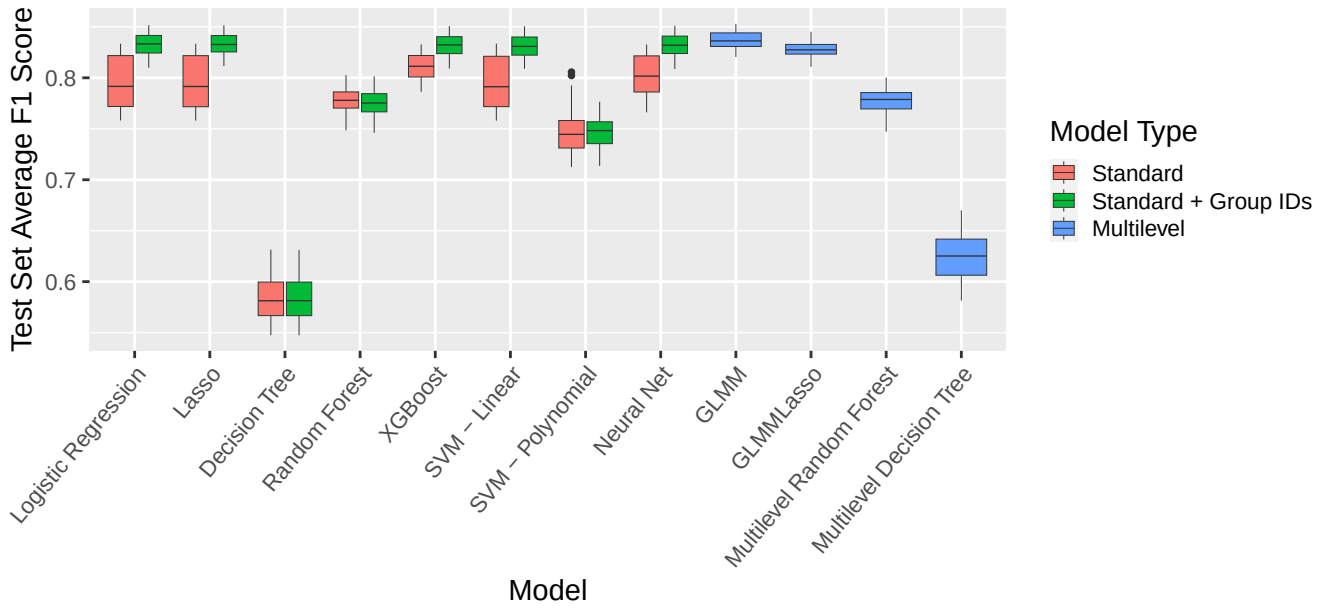

Figure A.22: Average test set F1 Score by model and model type.

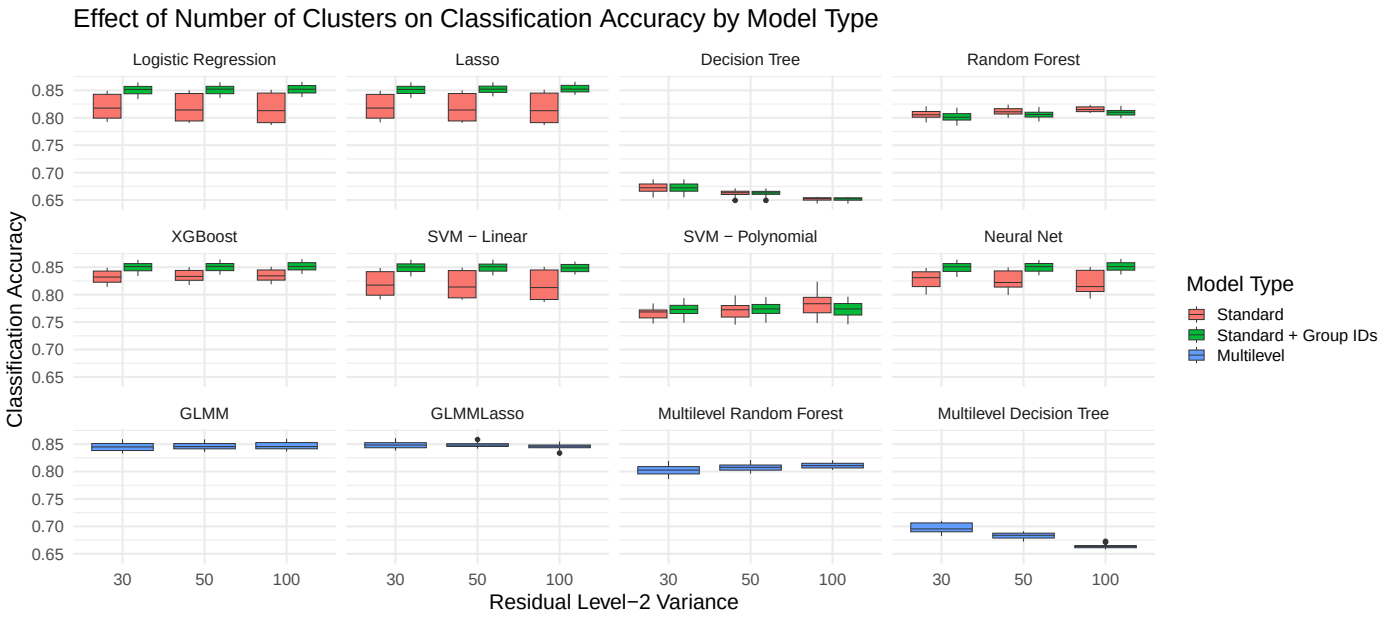

Figure A.23: Average test set accuracy by model, number of groups, and model type.

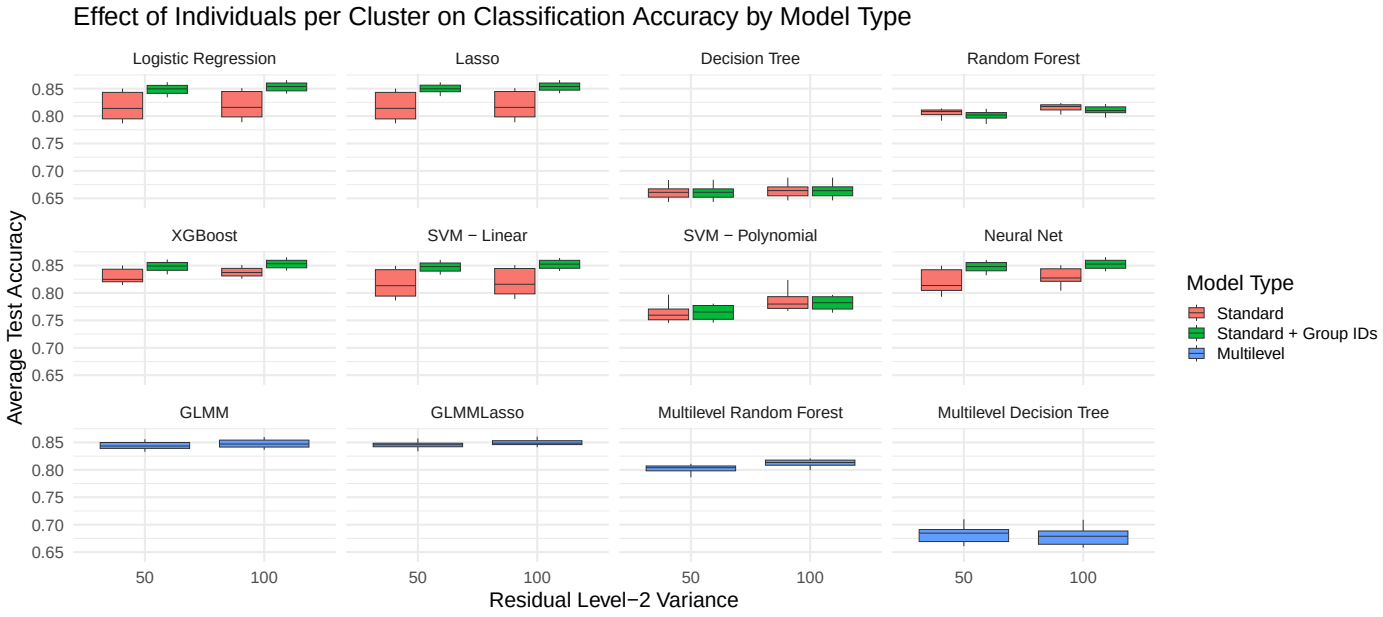

Figure A.24: Average test set accuracy by model, individuals per group, and model type.

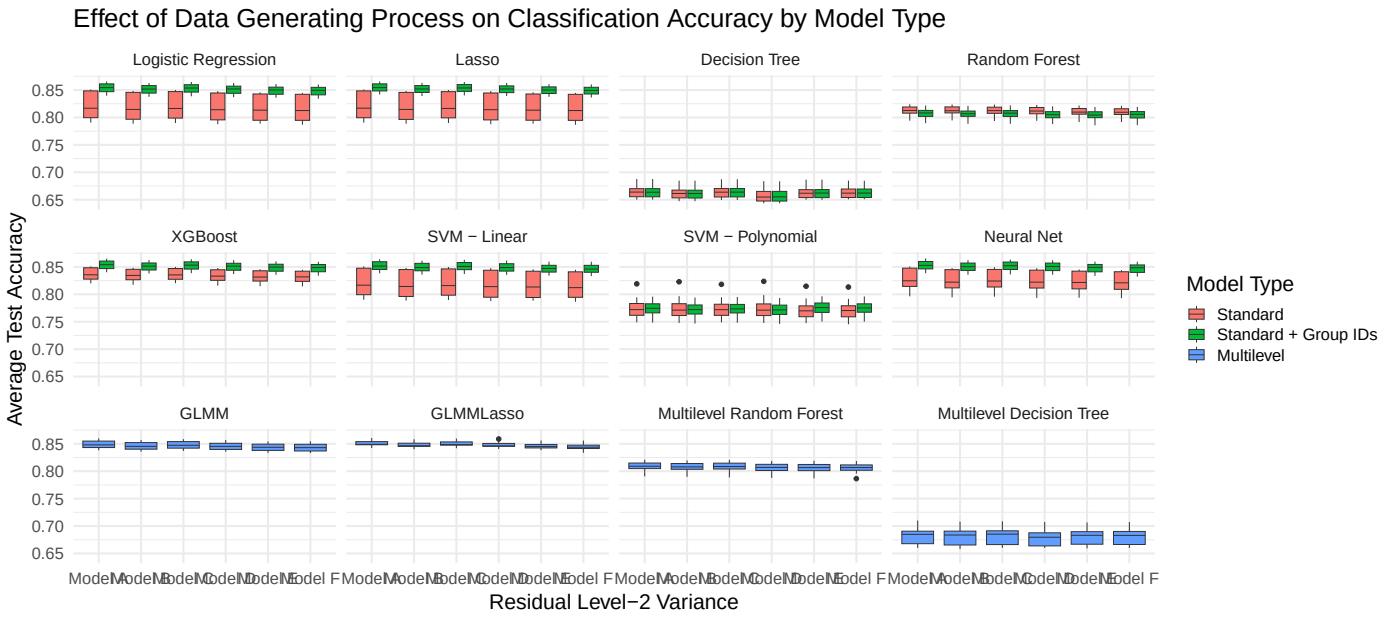

Figure A.25: Average test set accuracy by model, data generating process, and model type.

Bibliography

Aguiar, E., Chawla, N. V., Brockman, J., Ambrose, G. A., and Goodrich, V. (2014). Engagement vs performance: using electronic portfolios to predict first semester engineering student retention. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, LAK '14, pages 103–112, New York, NY, USA. Association for Computing Machinery.

Allison, P. (2008). Convergence Failures in Logistic Regression. *SAS Global Forum 2008*, 360.

Altabrawee, H., Ali, O. A. J., and Ajmi, S. Q. (2019). Predicting Students' Performance Using Machine Learning Techniques. *Journal of University of Babylon for Pure and Applied Sciences*, 27(1):194–205. Number: 1.

Arroway, P., Morgan, G., O'Keefe, M., and Yanosky, R. (2015). Learning analytics in higher education. Technical report, Research report. Louisville, CO: ECAR, March 2016. 2016 EDUCAUSE. CC by-nc-nd.

Bird, K. A., Castleman, B. L., Mabel, Z., and Song, Y. (2021). Bringing Transparency to Predictive Analytics: A Systematic Comparison of Predictive Modeling Methods in Higher Education. *AERA Open*, 7:23328584211037630. Publisher: SAGE Publications Inc.

BRIER, G. W. (1950). VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review*, 78(1):1 – 3. Place: Boston MA, USA Publisher: American Meteorological Society.

Cannistrà, M., Masci, C., Ieva, F., Agasisti, T., and Paganoni, A. (2020). Not the magic algorithm: modelling and early-predicting students dropout through machine learning and multilevel approach. *MOX-Modelling and Scientific Computing, Department of Mathematics, Politecnico di Milano, Via Bonardi*, pages 9–20.

Capitaine, L., Genuer, R., and Thiébaut, R. (2021). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, 30(1):166–184. Publisher: SAGE Publications Ltd STM.

Casella, G. and Berger, R. L. (2002). Statistical Inference . Pacific Grove, CA: Thomson Learning.

Chen, G., Taylor, P. A., Haller, S. P., Kircanski, K., Stoddard, J., Pine, D. S., Leibenluft, E., Brotman, M. A., and Cox, R. W. (2017). Intraclass correlation: Improved modeling approaches and applications for neuroimaging. *Human Brain Mapping*, 39(3):1187–1206.

Chen, J., Huang, K., Wang, F., and Wang, H. (2009). E-learning Behavior Analysis Based on Fuzzy Clustering. In *2009 Third International Conference on Genetic and Evolutionary Computing*, pages 863–866.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chen, Y., Johri, A., and Rangwala, H. (2018). Running out of STEM: a comparative study across STEM majors of college students at-risk of dropping out early. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 270–279, New York, NY, USA. Association for Computing Machinery.

Crane-Droesch, A. (2017). Semiparametric panel data models using neural networks. arXiv:1702.06512.

Elakia, G., Aarthi, N. J., and others (2014). Application of data mining in educational database for predicting behavioural patterns of the students. *Elakia et al,/(IJCSIT) International Journal of Computer Science and Information Technologies*, 5(3):4649–4652.

Engler, A. (2021). Enrollment algorithms are contributing to the crises of higher education.

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., and Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods*, 50(5):2016–2034.

Fontana, L., Masci, C., Ieva, F., and Paganoni, A. M. (2021). Performing Learning Analytics via Generalised Mixed-Effects Trees. *Data*, 6(7):74. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Gibson, D. C. and Ifenthaler, D. (2017). Preparing the next generation of education researchers for big data in higher education. *Big data and learning analytics in higher education: Current theory and practice*, pages 29–42. Publisher: Springer.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Gray, G., McGuinness, C., Owende, P., and Hofmann, M. (2016). Learning Factor Models of Students at Risk of Failing in the Early Stage of Tertiary Education. *Journal of Learning Analytics*, 3(2):330–372. Number: 2.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5):93:1–93:42.

Hadfield, J. D. (2010). MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. *Journal of Statistical Software*, 33:1–22.

Hajjem, A., Bellavance, F., and Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, 81(4):451–459.

Hajjem, A., Larocque, D., and Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics & Probability Letters*, 126:114–118.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

He, H. and Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Hedges, L. V. and Hedberg, E. C. (2007). Intraclass correlation values for planning group-randomized trials in education. *Educational Evaluation and Policy Analysis*, 29(1):60–87. Publisher: Sage Publications Sage CA: Los Angeles, CA.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. Publisher: Taylor & Francis.

Hox, J. (1998). Multilevel modeling: When and why. In *Classification, data analysis, and data highways: proceedings of the 21st Annual Conference of the Gesellschaft für Klassifikation eV, University of Potsdam, March 12–14, 1997*, pages 147–154. Springer.

Huang, H. H., Hsiao, C. K., and Huang, S. Y. (2010). Nonlinear Regression Analysis. In Peterson, P., Baker, E., and McGaw, B., editors, *International Encyclopedia of Education (Third Edition)*, pages 339–346. Elsevier, Oxford.

Huang, J. and Ling, C. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310.

Hutt, S., Gardener, M., Kamentz, D., Duckworth, A. L., and D'Mello, S. K. (2018). Prospectively predicting 4-year college graduation from student applications. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 280–289, New York, NY, USA. Association for Computing Machinery.

Huynh-Cam, T.-T., Chen, L.-S., and Le, H. (2021). Using Decision Trees and Random Forest Algorithms to Predict and Determine Factors Contributing to First-Year University Students' Learning Performance. *Algorithms*, 14(11):318. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3):685–695.

Jayaprakash, S. M., Moody, E. W., Lauría, E. J. M., Regan, J. R., and Baron, J. D. (2014). Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics*, 1(1):6–47. Number: 1.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27.

Kemper, L., Vorhoff, G., and Wigger, B. U. (2020). Predicting student dropout: A machine learning approach. *European Journal of Higher Education*, 10(1):28–47. Publisher: Routledge _eprint: https://doi.org/10.1080/21568235.2020.1718520.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Morgan Kaufman Publishing*.

KOTSIANTIS, S., PIERRAKEAS, C., and PINTELAS, P. (2004). Predicting Students' Performance in Distance Learning Using Machine Learning Techniques. *Applied Artificial Intelligence*, 18(5):411–426. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/08839510490442058.

Kotsiantis, S. B., Pierrakeas, C. J., and Pintelas, P. E. (2003). Preventing Student Dropout in Distance Learning Using Machine Learning Techniques. In Palade, V., Howlett, R. J., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, pages 267–274, Berlin, Heidelberg. Springer.

Kreft, I. G. and De Leeuw, J. (1998). *Introducing multilevel modeling*. Sage.

Lee, C.-A., Tzeng, J.-W., Huang, N.-F., and Su, Y.-S. (2021). Prediction of Student Performance in Massive Open Online Courses Using Deep Learning System Based on Learning Behaviors. *Educational Technology & Society*, 24(3):130–146. Publisher: International Forum of Educational Technology & Society ERIC Number: EJ1305779.

Lee, S. and Chung, J. Y. (2019). The Machine Learning-Based Dropout Early Warning System for Improving the Performance of Dropout Prediction. *Applied Sciences*, 9(15):3093. Number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

Li, G., Yang, M., Ran, L., and Jin, F. (2022). Classification prediction of early pulmonary nodes based on weighted gene correlation network analysis and machine learning. *Journal of Cancer Research and Clinical Oncology*.

Lin, S. and Luo, W. (2019). A New Multilevel CART Algorithm for Multilevel Data with Binary Outcomes. *Multivariate Behavioral Research*, 54(4):578–592.

Luan, H. and Tsai, C.-C. (2021). A review of using machine learning approaches for precision education. *Educational Technology & Society*, 24(1):250–266. Publisher: JSTOR.

Maas, C. J. and Hox, J. J. (2004). The influence of violations of assumptions on multilevel parameter estimates and their standard errors. *Computational Statistics & Data Analysis*, 46(3):427–440.

Maas, C. J. and Hox, J. J. (2005). Sufficient sample sizes for multilevel modeling. *Methodology*, 1(3):86–92. Publisher: Hogrefe & Huber Publishers.

Mangino, A. A. (2021). *Fixed Effects or Mixed Effects?: Predictive Classification in the Presence of Multilevel Data*. Ph.D., Ball State University, United States – Indiana. ISBN: 9798538117338.

Mangino, A. A., Bolin, J. H., and Finch, W. H. (2022). Fixed Effects or Mixed Effects Classifiers? Evidence From Simulated and Archival Data. *Educational and Psychological Measurement*, page 00131644221108180. Publisher: SAGE Publications Inc.

Mangino, A. A. and Finch, W. H. (2021). Prediction With Mixed Effects Models: A Monte Carlo Simulation Study. *Educational and Psychological Measurement*, 81(6):1118–1142. Publisher: SAGE Publications Inc.

Mayilvaganan, M. and Kalpanadevi, D. (2014). Comparison of classification techniques for predicting the performance of students academic environment. In *2014 International Conference on Communication and Network Technologies*, pages 113–118.

McArdle, J. J. and Ritschard, G. (2013). *Contemporary Issues in Exploratory Data Mining in the Behavioral Sciences*. Routledge. Google-Books-ID: OV9tAAAAQBAJ.

McNeish, D. and Kelley, K. (2019). Fixed effects models versus mixed effects models for clustered data: Reviewing the approaches, disentangling the differences, and making recommendations. *Psychological Methods*, 24(1):20–35. Place: US Publisher: American Psychological Association.

McNeish, D. and Stapleton, L. M. (2016). Modeling Clustered Data with Very Few Clusters. *Multivariate Behavioral Research*, 51(4):495–518. Publisher: Routledge _eprint: https://doi.org/10.1080/00273171.2016.1167008.

Meuleman, B. and Billiet, J. (2009). A Monte Carlo Sample Size Study: How Many Countries are Needed for Accurate Multilevel SEM? *Survey Research Methods*, 3:45–58.

Möller, A., George, A. C., and Groß, J. (2022). Predicting school transition rates in Austria with classification trees. *International Journal of Research & Method in Education*, 0(0):1–14. Publisher: Routledge _eprint: https://doi.org/10.1080/1743727X.2022.2128744.

Muschelli, J., Betz, J., and Varadhan, R. (2014). Chapter 7 - Binomial Regression in R. In Rao, M. B. and Rao, C. R., editors, *Handbook of Statistics*, volume 32 of *Computational Statistics with R*, pages 257–308. Elsevier.

Ngufor, C., Van Houten, H., Caffo, B. S., Shah, N. D., and McCoy, R. G. (2019). Mixed effect machine learning: A framework for predicting longitudinal change in hemoglobin A1c. *Journal of Biomedical Informatics*, 89:56–67.

Osmanbegović, E. and Suljic, M. (2012). DATA MINING APPROACH FOR PREDICTING STUDENT PERFORMANCE. *Journal of Economics & Business/Economic Review*, 10:3–12.

Pellagatti, M., Masci, C., Ieva, F., and Paganoni, A. M. (2021). Generalized mixed-effects random forest: A flexible approach to predict university student dropout. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 14(3):241–257. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sam.11505.

Philipp Kilham, Hartebrodt, C., and Kändler, G. (2019). Generating Tree-Level Harvest Predictions from Forest Inventories with Random Forests. *Forests*, 10.

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. _eprint: 2010.16061.

Rahman, M. M., Tabash, M. I., Salamzadeh, A., Abduli, S., and Rahaman, M. S. (2022). Sampling Techniques (Probability) for Quantitative Social Science Researchers: A Conceptual Guidelines with Examples. *SEEU Review*, 17(1):42–51.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Pearson, Boston Columbus Indianapolis New York San Francisco Upper Saddle River Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo, third edition, global edition edition.

Rutkowski, L., Rutkowski, D., and Svetina Valdivia, D. (2022). Multistage Test Design Considerations in International Large-Scale Assessments of Educational Achievement. In Nilsen, T., Stancel-Piatak, A., and Gustafsson, J.-E., editors, *International Handbook of Comparative Large-Scale Studies in Education: Perspectives, Methods and Findings*, pages 749–767. Springer International Publishing, Cham.

Sandra, L., Lumbangaol, F., and Matsuo, T. (2021). Machine Learning Algorithm to Predict Student's Performance: A Systematic Literature Review. *TEM Journal*, pages 1919–1927.

Schelldorfer, J., Meier, L., and Bühlmann, P. (2014). GLMMLasso: An Algorithm for High-Dimensional Generalized Linear Mixed Models Using 1-Penalization. *Journal of Computational and Graphical Statistics*, 23(2):460–477. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10618600.2013.773239.

Sela, R. J. and Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning*, 86(2):169–207.

Shahiri, A. M., Husain, W., and Rashid, N. A. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. *Procedia Computer Science*, 72:414–422.

Snijders, T. A. B. and Bosker, R. J. (2011). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling*. SAGE. Google-Books-ID: N1BQvcomDdQC.

Speiser, J. L. (2021). A random forest method with feature selection for developing medical prediction models with clustered and longitudinal data. *Journal of Biomedical Informatics*, 117:103763.

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., and Durkalski, V. L. (2019). BiMM forest: A random forest method for modeling clustered and longitudinal binary outcomes. *Chemometrics and Intelligent Laboratory Systems*, 185:122–134.

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., and Durkalski, V. L. (2020). BiMM tree: a decision tree method for modeling clustered and longitudinal binary outcomes. *Communications in Statistics - Simulation and Computation*, 49(4):1004–1023. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03610918.2018.1490429.

Thompson, N., Greenewald, K., Lee, K., and Manso, G. F. (2023). The Computational Limits of Deep Learning. In *Ninth Computing within Limits 2023*, Virtual. LIMITS.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288. Publisher: [Royal Statistical Society, Wiley].

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259. Publisher: Elsevier.

Wu, C., Barczyk, A. N., Craddock, R. C., Harari, G. M., Thomaz, E., Shumake, J. D., Beevers, C. G., Gosling, S. D., and Schnyer, D. M. (2021). Improving prediction of real-time loneliness and companionship type using geosocial features of personal smartphone data. *Smart Health*, 20:100180.

Zou, H. and Hastie, T. (2005). Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Statist Soc B. 2005;67(2):301-20. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:301 – 320.