
MASTER THESIS

# Optimizing Mixed-Effect Models for Improved Performance and Interpretability

*Author:*
**Harshal Vijaykumar Talsania**
*Matriculation Number:* **220202838**

*Supervisor:*
**Jun.-Prof. Dr. rer. nat.**
**Martin Becker**

*Co-supervisor:*
**Jun.-Prof. Dr.**
**Stefan Lüdtke**

*A thesis submitted in fulfillment of the requirements*
*for the degree of Master of Science (M.Sc.)*
*in the*
**Computational Science and Engineering**
10 April 2024

**Universität Rostock**
Fakultät für Informatik und Elektrotechnik
Institut für Visual and Analytic Computing
Intelligent Data Analytics

# Declaration of Authorship

I, **Harshal Vijaykumar Talsania**, declare that this thesis titled, "Optimizing Mixed-Effect Models for Improved Performance and Interpretability" and the work presented in it are my own. I confirm that:

- This work has been accomplished solely or mainly while in candidature for a master degree at the Rostock University.
- This work has been carried out under the guidance of **Jun.-Prof. Dr. rer. nat. Martin Becker** and co-supervisor **Prof. Stefan Lüdtke**.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, which is always clearly attributed.
- Where I have quoted from the work of others, the source is always provided. With the exception of such quotations, this thesis is entirely my own work. As well, I have marked verbatim and indirect quotations as such.
- I have acknowledged all the main sources of help including ChatGPT by OpenAI, for suggestions for structure, and general guidance on academic writing. The analytical and intellectual content of this thesis, however, is solely my contribution and reflects my understanding of the subject.

Signed 

Date and Place: 10 April 2024, Rostock

# Abstract

Mixed effect models allow for explicit modelling of different relationships between features and outcomes within subsets of the data. Such subsets are defined by group variables. However, not each group necessarily has its own distinct set of relationships. Certain groups may share some kind of relationship and might not contribute much unique information, making them redundant with regard to model fitting.

If the number of groups increases for the same size of the dataset, two problems arise. Firstly, the task of fitting good relationships for each group may become challenging for existing algorithms, as the number of samples per group decreases. Secondly, even if they can form good relationships, it might get increasingly difficult to interpret that many groups in a meaningful way.

In order to tackle these problems, we analyse two key aspects. First, how the model performance of a mixed-effect model changes with an increasing number of groups within the dataset. Second, how challenging the interpretability gets. Our central part of this work is to develop methods to reduce the number of groups. Subsequently, we conduct a comprehensive analysis, beginning with an evaluation of the impact of a method on model performance of a mixed-effect model. Finally, we examine whether these methods enhance interpretability or introduce a trade-off between interpretability and model performance of a mixed-effect model.

For performing the discussed approaches, various experiments will be conducted on synthetically generated datasets. Existing mixed-effect models will be used to derive model performance. We will then evaluate the developed methods and the interpretability of synthetic data. Moreover, we may explore the application of these methods on real-world data to further validate their practical utility.

# *Acknowledgements*

I would like to express my deepest gratitude to **Jun.-Prof. Dr. rer. nat. Martin Becker** and co-supervisor **Prof. Stefan Lüdtke** for allowing me to work with this interesting and challenging theme. I am sincerely thankful for being patient with me and being available to always answer all my questions.

I am also deeply grateful to my dear parents for their support and encouragement, constant sincere efforts and companionship during challenging and difficult moments.

Lastly, thanks to my guru, and friends for their moral and emotional support. Their presence and encouragement have made this journey more enjoyable and fulfilling.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

**In the field of data analysis, one would be interested in understanding the relationships between features.** Conventional methods like linear regression provide valuable explanations like feature coefficients, but they are not always suitable for handling complicated data structures, particularly when working with categorical features.

**Consider researching the rates of plant development in several gardens,** where each garden has its unique soil composition and watering schedule. Here, the growth rate of each plant is not just affected by factors like sunlight and fertilizer used, but also by the specific garden in which it is grown. It is complicated to figure out why one garden is better or different than others.

**Even if one can fit the data effectively,** one may not be able to determine which feature, either by itself or in conjunction with other features, plays the most important role. This is where mixed-effect models come into play. They are also known as Multilevel Models or Hierarchical Linear Models [1]. They provide an effective statistical tool for navigating the complex interactions between fixed and random factors, refer to section-2.1.

**Mixed-effect models are currently being used in a variety of studies,** such as those on educational research, psychological applications, ecological research and disease detection [2][3]. In this study, the main focus is to understand the mixed effects models, their limitations, and the solutions to improve them.

## 1.1 Motivation

**The complexity and volume of data across diverse fields are constantly increasing.** To analyse data is now not just to get better predictions but also to understand the reason behind it and find unanswered questions. To achieve that the demand for robust and flexible models is increasing.

**Mixed-effects models (MEMs) are proven to be great statistical tools and are helpful in many research areas.** They stand out for their ability to handle nested data structures, which are common in fields like medicine and finance. The data from these fields are quite complex and there are correlations between classes. Handling those relationships with conventional models leads to poor performance and interpretability. Mixed-effect Models on the other side, provide much more accurate estimates by handling those complex relationships [4]. In the medical and ecological domains, mixed-effects models enhance understanding by accounting for individual variability and spatial hierarchies, respectively, providing insights inaccessible through simpler methods [5].

**The application of those models is increasing and is now being considered in a much more important domain which is genomics studies.** Mixed-effect models can be used to understand the dynamics of gene regulation and identify the chain of genes responsible for a specific disease [6][7]. Unlike other models, mixed-effects models can distinguish variability due to random effects of individuals and fixed effects which is a treatment they are getting, that helps to understand the underlying biological processes, which ultimately improves interpretation [8].

**Mixed-effects models are flexible over typical data analysis,** especially when dealing with large categorical features. These models can easily handle high cardinality, which is common in most fields. Even with more complex structures when these features are nested or consist of some correlation among instances in the same cluster, these models can tackle large datasets with high cardinality over typical analysis methods [9].

## 1.2 Problem Definition

**Mixed-effect models are usually considered complex models** because they take the combination of fixed and random effects into account. Identifying the nature of random effects that the data inherits is also crucial. The study aims to identify the problem for a large number of groups.

**When it comes to high-cardinality categorical features,** each category has a small amount of subsets. If there are a smaller number of categories then one can handle the problem with either one-hot encoding or other simpler approaches but still, there will be a possibility to fit bad relationships when the data has a complex structure [10]. Furthermore, not every group from a large number of groups has a unique relationship with other features. Certain groups may share similar kinds of relationships and can be treated as one. In that scenario, the mixed-effect models might face difficulty in fitting the data in the correct way. Not all models with higher prediction accuracy also provide better interpretation. The number of groups included in mixed-effect models increases the complexity of the interpretation. Evaluating the relation and importance of a group to other features is what makes the interpretation more challenging.

**The major goal of this thesis is to investigate the behaviour of mixed-effects models with a high number of categorical features**, as well as approaches for reducing those categorical features to overcome the model's limitations. The approaches will be evaluated for model performance and interpretation to determine whether they may be improved. The study mainly focused on the regression setting and linear mixed-effect models.

## 1.3 Structure of Work

The structure of this thesis is organized as follows: Chapter 2 introduces the foundational concepts of mixed-effect models, detailing the differences between fixed and random effects, and exploring various types of random effects. Chapter 3 is dedicated to a review of recent studies and efforts aimed at addressing the

problem with high-cardinality categorical features. Chapter 4 discusses the variety of methods and approaches developed to manage the high-cardinality categorical features. Chapter 5 explains the process used for generating synthetic data, which is utilized in conducting numerous experiments. Chapter 6 presents the experimental results, demonstrating the impact of a large number of groups on model performance and the application of methods on real-world datasets. Chapter 7 discusses the efficacy of newly developed methods for group reduction, their limitations, and the interpretability aspects of the methods developed. Finally, Chapter 8 provides a concise summary of the entire study and discusses possible future research in this field.

# Chapter 2

# Fundamentals

**In this chapter, we give a general idea about the mixed-effects models,** by explaining what the mixed effects are. Then we talk about the model that is designed to make good predictions on data that inherits mixed effects. Furthermore, we will also see different types of mixed effects and how the linear mixed-effect models are adept at handling those types, using their mathematical formulation.

## 2.1 What is Mixed-Effects?

**In the mixed-effect models, the term mixed-effect is simply a combination of both fixed effects and random effects.** By combining these effects, models can analyse complex data with both constant and variable components across different groups.

1. **Fixed Effects:**

   **The fixed effects are parameters that have the consistent influence of an independent variable on the dependent variable across all individuals.** These effects represent population-level effects. For instance, consider a study analyzing the impact of seasonal variations on air-conditioner sales. The increase in sales during the summer season would be a fixed effect, applicable to all retailers, whether they operate in India or Germany. This effect remains constant across the population. If a model considers only fixed effects, the entire data will have only one intercept and slope [2], as shown in the figure-2.1,

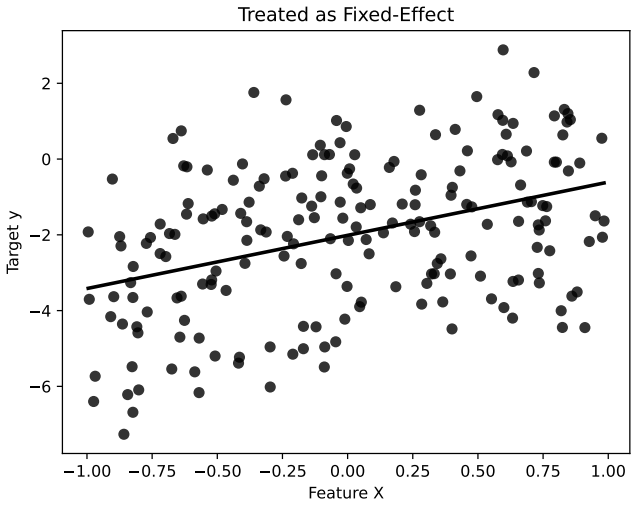



FIGURE 2.1: Feature treated as a fixed-effect

2. **Random Effects:**

**The random effects represent the effects of variables that have a varying effect on the outcome variable across groups or individuals.** These effects are unique to individual groups and allow for individual differences, which are typically available in complex clustered data structures. In a previous example of air-conditioner sales, individual countries can be considered as random effects. This is because each country has unexplained variations in sales, which are not adequately captured by the fixed-effect model alone. When a model explicitly accounts for random effects, then it allows each group to have a separate intercept or slope [2]. Consider figure-2.2,

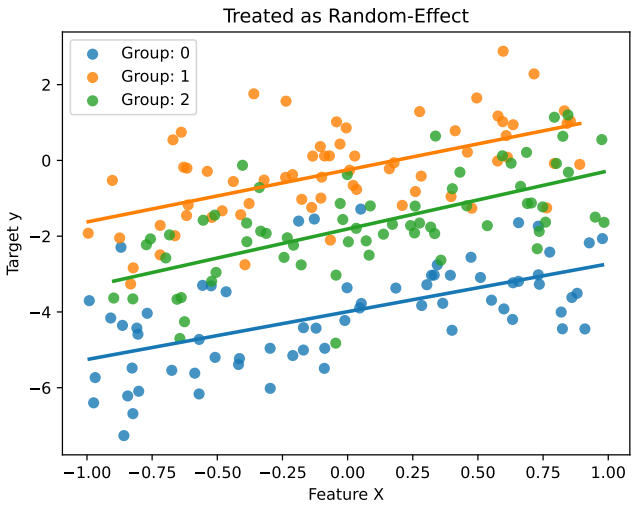



FIGURE 2.2: Feature treated as a random-effect

## 2.2 Linear Mixed-Effects Models

**Linear mixed-effects models, an advanced form of the general linear model, are designed to handle complex data structures by incorporating both fixed and random effects [11].** These models effectively handle nested or hierarchical data structures, taking into account the influence of nested associations on observations [12]. These models prove to be especially beneficial in scenarios where multiple observations are collected from the same units repeatedly, or when data is clustered from groups of interconnected subgroups [2].

**To obtain the general formulation of a linear mixed-effect model,** let's start with the basic linear regression formula and extend it to add the random effects term, [13], [14].

The basic linear regression formula:

$$Y = X\beta + \epsilon \tag{2.1}$$

Let's assume a dataset with $n$ instances. Where equation 2.1 denotes,

- $Y$, the dependent variable vector, $n \times 1$.
- $X$, the design matrix for fixed effects for $p$ number of predictors, $n \times p$.
- $\beta$, the vector of fixed effects coefficients, $p \times 1$.
- $\epsilon$, the vector of residual errors, $n \times 1$.

Now, to incorporate random effects, another term $Zu$ is added,

$$y = X\beta + Zu + \epsilon \tag{2.2}$$

Where equation 2.2 denotes,

- $Z$ is the design matrix for random effects with $q$ number of groups, $n \times q$.
- $u$ is the vector of random effects coefficients, $q \times 1$.

Above equation 2.2 is the general formula for a mixed-effects model. The formulation combines both fixed and random effects, allowing for the modelling of both population-level and group-level variations within the dataset.

## 2.3 Types of Random Effects Models

**Random effects can be primarily incorporated into models in three special cases.** Consider the figure-2.3,

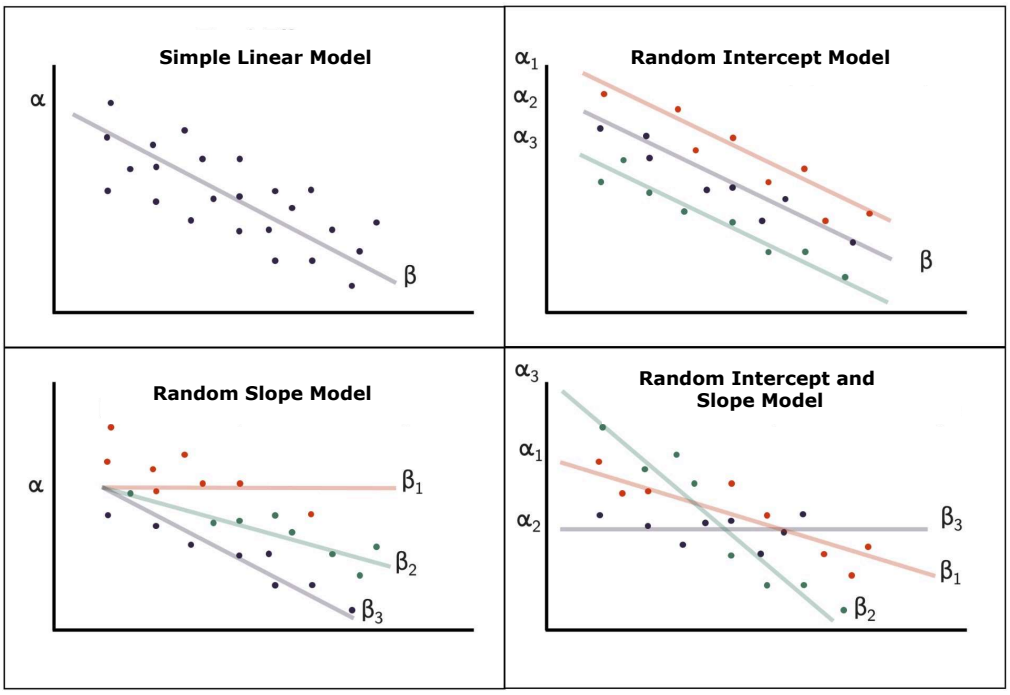



FIGURE 2.3: Types of Random Effects model [15]

The figure-2.3 consists of three random effects along with a simple linear model, which has a single intercept($\alpha$) and slope($\beta$). In random effects, data is shown with three categories that have separate regression lines. The random effects models can be derived from the general formulation of MEMs. By breaking down the random effects term "Zu" from eq. 2.2, we get three special cases:

1. **Random Intercept Model:**

   **In a random intercept model, each group is allowed to have its own intercept along with a global intercept of the model.** As we can see in the figure-2.3 the name suggests, the intercepts$(\alpha_1, \alpha_2, \alpha_3)$ vary with a constant slope$(\beta)$.

$$y_{ij} = \beta X_{ij} + u_{0j} + \varepsilon_{ij} + \epsilon \tag{2.3}$$

   The equation 2.3 can be derived from the general mixed-effects form equation 2.2. Here, $u_{0j}$ is the random intercept for the $j$-th group, which is part of the $Zu$ term in the general model and $\varepsilon_{ij}$ is the random effect error term associated with the $j$-th group.

   **Example:** In the education domain, students (i) within a school (j) may have different grades in annual results but they study at the same rate (slope) when given a specific educational environment.

   In the figure 2.3 with the random intercept model, assume the grades/results of three schools shown with separate regression lines and all three schools have separate intercepts but a constant slope.

2. **Random Slope Model:**

   **Similarly in a random slope model, each group is allowed to have its slope.** As we can see in the figure-2.3 the name suggests, the slopes$(\beta_1, \beta_2, \beta_3)$ vary by keeping the intercept$(\alpha)$ constant.

$$y_{ij} = \beta X_{ij} + u_{1j} X_{ij} + \varepsilon_{ij} + \epsilon \tag{2.4}$$

   The equation 2.4 can be derived from the general mixed-effects form equation 2.2 by allowing the term $Zu$ to include $u_{1j} X_{ij}$, where the term $u_{1j}$ is the random effect associated with the slope of $j$-th group and $\varepsilon_{ij}$ is the random effect error for the $j$-th group.

   **Example:** In a hospital, different patients (i) may respond differently to the medication (j) for a specific disease, so the effect of the medication (slope) may vary by patient.

In figure 2.3 with the random slope model, assume the response of different medications shown separately with three regression lines. One medication recovers patients at a faster rate, while with the second medication, the patients take time to recover. Here, the rate of recovery (slope) might vary with different medications.

3. **Random Intercept and Slope Model:**

   **In this model, each group is allowed to have its intercept as well as slope.** As we can see in the figure-2.3 the name suggests, both intercepts - $(\alpha_1, \alpha_2, \alpha_3)$ and slopes$(\beta_1, \beta_2, \beta_3)$ vary at different rate.

$$y_{ij} = \beta X_{ij} + u_{0j} + u_{1j} X_{ij} + \varepsilon_{ij} + \epsilon \tag{2.5}$$

   The equation 2.5 can be derived from the general mixed-effects form equation 2.2 by allowing the term $Zu$ to include both terms $u_{0j}$ & $u_{1j} X_{ij}$, where $u_{0j}$ is random intercept and $u_{1j}$ is random slope for $j$-th group and $\varepsilon_{ij}$ is the random effect error term associated with the $j$-th group.

   **Example:** In a weight loss study, 3 participants follow the same diet, but they may start following the diet at different weights and lose weight at different rates.

   In figure-2.3 with Random Intercept and Slope model, assume the journey of weight reduction of three participants is shown. All three had different weights (intercepts) when they started but in the end, they lost their weight at different rates (slopes).

# Chapter 3

# Related Work

As the study aims to solve the issue of high-cardinality in data, this section discusses the previous approaches that handle the high-cardinality categorical feature. Specifically, we review deep learning and tree-based models. We mainly discuss how these models tackle high cardinality to improve performance and interpretability.

**In the analysis of high-cardinal data,** traditional fixed-effects models may not fully capture complexities within clustered or hierarchically structured datasets. A significant study suggests that explicitly treating categorical features as random effects improves performance. By introducing cluster-level random effects, the model effectively handles the correlations within clusters, leading to better estimation [16]. The researchers developed several approaches to handle complex data with high cardinality to improve performance and interpretability. Here, we have considered mainly two kinds of approaches:

**By combining deep learning with mixed effects,** the model LMMNN [17] incorporates random intercepts and random slopes in regression settings. LMMNN introduced a negative log-likelihood loss function used in LMM (Linear Mixed Models) with SGD (Stochastic Gradient Decent) to avoid issues with overfitting and scalability [18]. However, LMMNN's random slope modelling is limited to longitudinal data with repeated measures, as it uses a separate temporal variable to model random slopes. Another model, DeepGLMM [19], also addresses random slopes theoretically but focuses solely on random intercepts [20]. ARMED [20] on the other hand, uses a separate random effects subnetwork for a domain adversarial classifier to introduce random effects into a dense feedforward neural network. ARMED mainly focused on improving prediction on unseen clusters or

categories and interpretability. The ARMED model demonstrates superior performance in handling various types of random effects compared to other deep neural network models. However, it presents challenges, including complex hyperparameter settings, lengthy training times, and a primary focus on binary classification tasks. Notably, the authors evaluated ARMED using data with only 20 categories, which does not solve the problem of data with high cardinality [20].

**The tree-boosting approaches** were also developed to handle random effects with high cardinality. One of the studies [21] presents an integration of mixed effects within the random forest framework called MERF [21] to handle clustered data. MERF has also shown better performance on high-cardinal data with compared to a simple random forest algorithm. However, MERF does not explicitly specify whether random effect features are treated as random intercepts, slopes, or both. This lack of clarity limits the model's interpretability and practical application. The interesting fact was that tree-boosting algorithm like GPBoost with random effects outperforms deep neural networks with random effects approaches on tabular data [10].

**Interpretability.** There have been many approaches that work on improving performance but due to the complexity of mixed-effect models, the interpretability is still a big problem to be solved. For instance, LMMNN, due to their integration with deep neural networks, does not provide any insights regarding interpretability [20]. In contrast, ARMED improves interpretability by separating random and fixed effects into distinct subnetworks, allowing for a better understanding of the impact of different clusters on the data. However, it is only applicable for a single level of random effects [20]. MERF on the other hand, does not provide any information to interpret the results and it assumes that the random effects have a linear relationship [21]. The model MixedLM from Python statsmodels is also able to provide general random effects coefficients. However, it is numerically challenging, when the random effects covariance matrix is singular [22].

**Strategy.** Our approach for addressing these issues by simply to somehow reduce the number of groups. The strategy is to focus on reducing the number of groups within the dataset. By merging similar groups, we anticipate minimal impact on model performance and lesser groups are comparatively more feasible to interpret than a higher number of groups. With this approach, we are aiming to improve the interpretability. Understanding the contribution of features

to any decision on why these groups are combined for reduction helps improve interpretability.

# Chapter 4

# Methodology

**In our methodology, we adopt a central strategy to effectively handle the high-cardinality group feature by reducing the number of groups and clustering them to treat them as a unified entity.** The chapter introduces several approaches and methods to facilitate group reduction. Recognizing the limitation of the brute force approach, we have suggested heuristic approaches to identify similar subsets of groups. Subsequently, we employ various methods to cluster groups based on their unique characteristics. Our goal is to combine groups in a manner that optimizes performance while also explaining their grouping.

## 4.1 Brute Force Approach

The algorithm exhaustively evaluates all possible combinations to search for optimal subsets from a set of unique groups available in a categorical feature.

The generation of all possible combinations within a set utilises the concept of Bell numbers [23]. However, certain criteria must be satisfied for evaluation:

- All subsets must be derived from the original set.
- The intersection of any two distinct subsets must be a null set.
- The number of subsets must be greater than one and less than the length of the original set.

For instance, considering the example in figure-4.1, where the set contains three unique groups, the resulting combinations are:

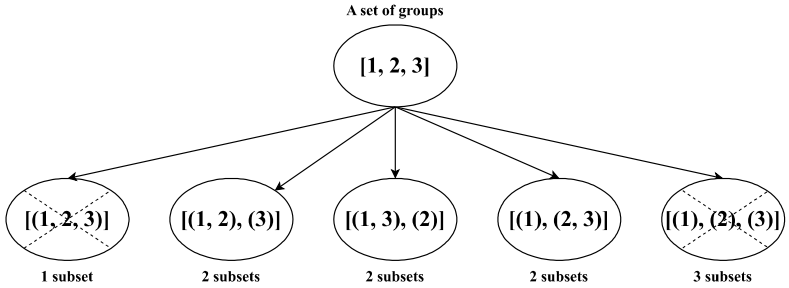



FIGURE 4.1: Bell numbers algorithm provides 5 combinations, although first and last combinations are always excluded as they violate the third criteria.

**The problem with the brute force approach** is the exponential increase in possible combinations with the addition of each group. The overwhelming number of combinations is computationally expensive for a large number of groups, see table 4.1. This approach not only enumerates a massive number of combinations but also the model used to evaluate each combination may take a longer time if a complex model like a neural network is used. The approach also lacks interpretability, as it fails to provide explanations for the chosen group combinations.

| **Unique Groups** | **5** | **10** | **15** | **20** | **100** |
|---|---|---|---|---|---|
| **Possible Combinations** | 50 | 115973 | $\approx 1.383 \times 10^9$ | $\approx 5.172 \times 10^{13}$ | $\approx 4.759 \times 10^{79}$ |

TABLE 4.1: Exponential increase in possible combinations

## 4.2 Heuristic Approaches

**Recognizing the computational limitation of the brute force approach,** heuristic methods employ a more strategic search strategy. Instead of searching the entire search space to find a combination with optimal subsets, the methods presented here follow the strategy of backward elimination. The algorithm iteratively removes an element from the initial set of groups that are unlikely to contribute to an optimal solution.

For a better explanation, consider the algorithm given below:

**Pseudocode**

**Result**: Best Performed Subsets
**Input**: Set of single group elements, **IA**:= Initial set of Groups

**BS**:= Best subset from group elements
**R**:= Remaining group elements *# Group elements, which did not make into BS*
**CA**:= Current array *# array under evaluation*
**CA_results**: = Subset-mse results *# To choose best subset with lowest mse*

```
1.  start
2.  BS = [ ], R = None, CA = IA, # Initialize parameters
3.  while R is None do
4.          CA_results = [ ]
5.          while the length of CA > 2 do
6.                  # function finds one element, that does not belong with CA
7.                  (subset, error) = RemoveOneBadGroupElement(CA)
8.                  insert (subset, error) into CA_results
9.                  # Update CA to repeat the same, as one element is removed
10.                 update CA = subset
11.         endwhile
12.         insert min(CA_results) into BS # subset with lowest error
13.         # Rest of the removed group elements are combined
14.         CA = CombineRemovedGroupElements()
15.         if the length of CA <= 2 then R = CA endif
16. endwhile
17. # Further Evaluation of R with BS
18. # Compute performance of all possible subset combinations to add R into BS
19. if error of "R – BS" < error of "BS" then return R – BS
20. else return BS endif
21. End
```

FIGURE 4.2: Heuristic Approach Algorithm

1. **Input/Output and Parameters**

   **Input:** IA = Initial array, The unique groups from the categorical feature in the form of an array. Here consider it as a set.

   **Output:** Best performed subsets derived from an initial array IA.

   **Parameters:**

   (a) **BS (Best Subset):** Starts as an empty list created to store a subset found with the lowest error(MSE).

   (b) **R (Remaining Group Elements):** Initially set to None, later to include group elements not included in BS.

   (c) **CA (Current Array):** Initially, it is assign as IA. It represents the combination that is currently being evaluated.

   (d) **CA_results (Current Array Results):** An empty list, to temporarily hold the performance results for different CAs.

2. **Main Evaluation While Loop**
   The main loop begins with the initial parameters. CA_results is reset to an empty list.

   (a) **Main Outer Loop Condition**
   Continues until R is no longer None, indicating there are no more elements to evaluate.

   (b) **Inner While Loop**

      i. **Inner Loop Condition:** Continues as long as CA has more than two elements, suggesting elements can still be removed from CA.

      ii. **Working:** The elements are iteratively removed one by one from CA using one of the methods described later in Section 4.2.1 and 4.2.2, here it is stated as function, $RemoveOneBadElement(CA)$. The result is stored in CA_results and CA is updated to the new, reduced array.

      iii. **Inner loop Termination Condition:** The inner loop ends when the length of CA is greater than 2.

   (c) **Finding BS from CA_results**
   After the inner loop, the subset with the lowest error from CA_results, is selected and stored in BS. Removed elements are set aside for combining again.

   (d) **Combining Removed Elements**
   The removed groups are combined again with a function, here it is stated as, $CombineRemovedGroupElements()$. The function output returns an array which will be assigned to CA for iterating again in the inner while loop.

   (e) **Outer loop Termination Condition**
   If CA's length is less or equal to 2, then array elements are assigned as remaining group elements in R. Otherwise, keep CA updated for the next loop iteration.

   At this phase, BS are the subsets created from a set, which has the best performance yet.

3. **Evaluation of Remaining Group Elements R with BS**
   The combinations of R with BS will be formed for further evaluation. The

strategy is to see if the performance of the combination can be further increased. The integrated elements of R into existing BS subsets are denoted as the R-BS combinations.

All the formed combinations will be evaluated and the one with the best performance among them will be selected for comparison.

4. **Final Comparison of R-BS with previous BS**
   The final condition checks if a newly found R-BS combination outperforms the original BS with a lower error. According to that condition, the algorithm decides its output, whether it is BS or R-BS.

The function $RemoveOneBadElement(CA)$ plays an important role in the algorithm. Based on that the algorithm finds the combination of subsets, which has better performance. We have introduced two methods for implementing backward elimination. Refer to section-4.2.1 and 4.2.2.

**Overall, the heuristic approach provides faster results than the brute-force approach.** Also, it uses informed methods to guide the algorithm. This approach may not always identify the absolute optimal solution, they prioritize computational efficiency by converging on good enough solutions.

### 4.2.1 Variance-based Group Removal

**This method focuses** on minimizing the variance of within-subset and maximizing the variance of between-subset. To calculate the variance of subsets, any measure which defines the group feature in one value can be considered. For instance, we have considered the Shapley value of groups discussed in section 4.3.2.

**To exclude a group from a subset,** the method iteratively removes each group element, calculates the variance of remaining groups in a subset and selects the group leading to the subset with the lowest variance. This can better be understood with an example.

Let us assume, the data has 5 groups and the corresponding Shapley values (refer section 4.3.2) are:

Set of groups = (1,2,3,4,5)

| **Group No.** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Shapley Value** | 2.5 | 9.1 | 11.2 | 3 | 7.5 |

| **Group Excluding** | **Sum of Squared Differences** |
|---|---|
| (1,2,3,4) (5) | 57.09 |
| (1,2,3,5) (4) | 41.23 |
| **(1,2,4,5) (3)** | **32.21** |
| (1,3,4,5) (2) | 50.53 |
| (2,3,4,5) (1) | 36.34 |

TABLE 4.2: Variance-based Group Removal: From the shapley values of groups, the sum of squared differences is calculated for the possible combination.

From the table-4.2, the method identifies Group - "3" for removal because it has the lowest sum of squared differences equivalent to the lowest variance.

This selection occurs before evaluating the performance (MSE) of combinations. As a result, the method significantly reduces the time required to train the model by eliminating less informative groups. This selection strategy accelerates the process compared to the brute force approach, which evaluates all possible combinations.

### 4.2.2 Performance-based Group Removal

**This method evaluates the impact of each group on the overall performance of the model.** Unlike variance-based group removal, which considers the internal variability within a group, performance-based removal identifies groups that contribute less significantly to the model performance.

**Similar to variance-based removal,** the method iteratively removes groups one at a time from a subset. However, instead of analyzing group variance, it evaluates the performance of the remaining subset after each removal. The group that leads to the highest model performance (or lowest MSE) will be removed.

**The effect of the performance-based group removal method** is highly dependent upon a model used for evaluation. This is because when a suitable model is

selected, the method can efficiently identify which group should be eliminated.

An example below illustrates the functioning of the method:

Let us assume, the data has 5 groups

Set of groups = (1,2,3,4,5)

| Group Excluding | Model Performance (MSE) |
|---|---|
| (1,2,3,4) (5) | 3.452 |
| **(1,2,3,5) (4)** | **1.459** |
| (1,2,4,5) (3) | 6.897 |
| (1,3,4,5) (4) | 11.598 |
| (2,3,4,5) (5) | 20.100 |

TABLE 4.3: Performance-based Group Removal: The method derives model performance for each possible combination to find the one to exclude.

In table-4.3, for each time the group element is eliminated, the grouping feature needs to be relabeled and data needs to be retrained on the model. That consumes more time than the variance-based method. The method greedily searches for the combination of subsets with the highest model performance, which may provide a better solution than the variance-based method.

## 4.3 Based on Unique Characteristics

**In this novel approach, we aim to cluster groups solely based on their unique attributes.** In complex datasets, unique characteristics can vary such as feature importance, regression coefficients, or any other defining parameters. Our methods involve identifying these distinct characteristics for each group and employing clustering techniques to combine similar groups.

For clustering methods, we have considered:

- **Jenks Natural Breaks**: The algorithm offers a data clustering technique specifically aimed to optimize the classification of values of 1-dimensional

data. It is particularly used when dealing with data that already exhibit inherent groupings, which relates to the term "Natural Breaks" [24].

This algorithm is particularly advantageous when data already exhibit inherent groupings. In these methods, after extracting a single unique measure per group, the primary objective is to group similar measure values and consider it a cluster.

- **K-Means Clustering**: It is a centroid-based clustering algorithm used to cluster the data with multiple dimensions. It divides the N-dimensional data into k clusters. This method allocates data points iteratively to the nearest cluster centroid, reducing within-cluster variation while increasing between-cluster variance [25].

**In order to identify these unique characteristics of a group,** we explore two distinct methods. Firstly, we utilise Python's MixedLM mixed-effects model, providing the random effects coefficients for both the intercept and slope of groups. Secondly, we compute shapley values for each group, which quantify their individual contributions to the dataset.

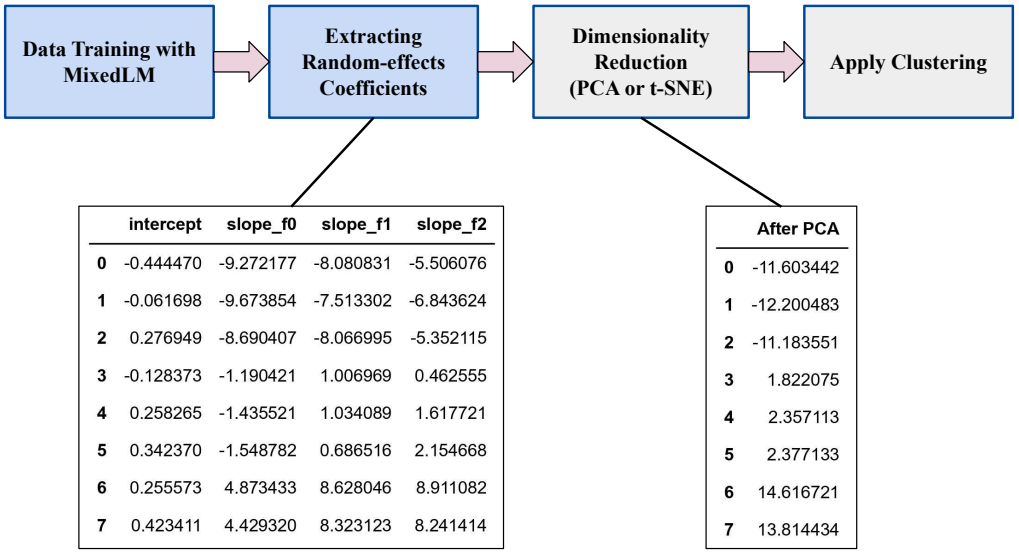
### 4.3.1 Extracting MixedLM random effects coefficients

**The method uses the existing mixed-effect model MixedLM.** The advantage of this model is, that it interprets random effects coefficients and provides parametric values for both the intercept and slope [22].

**Additionally, the model is capable of handling multi-dimensional data.** This is particularly advantageous as it allows for the calculation of slopes associated with predictor features, providing unique characteristics of individual groups.

The method has been divided into 4 steps elaborated below:

1. **Data Training:** The initial step involves training the entire dataset. This allows the model to capture the overall structure and relationships within the data. Ensure preprocessing of the data, which also includes identifying required fixed predictors and a random effects grouping feature.




| | intercept | slope_f0 | slope_f1 | slope_f2 |
|---|---|---|---|---|
| 0 | -0.444470 | -9.272177 | -8.080831 | -5.506076 |
| 1 | -0.061698 | -9.673854 | -7.513302 | -6.843624 |
| 2 | 0.276949 | -8.690407 | -8.066995 | -5.352115 |
| 3 | -0.128373 | -1.190421 | 1.006969 | 0.462555 |
| 4 | 0.258265 | -1.435521 | 1.034089 | 1.617721 |
| 5 | 0.342370 | -1.548782 | 0.686516 | 2.154668 |
| 6 | 0.255573 | 4.873433 | 8.628046 | 8.911082 |
| 7 | 0.423411 | 4.429320 | 8.323123 | 8.241414 |

| | After PCA |
|---|---|
| 0 | -11.603442 |
| 1 | -12.200483 |
| 2 | -11.183551 |
| 3 | 1.822075 |
| 4 | 2.357113 |
| 5 | 2.377133 |
| 6 | 14.616721 |
| 7 | 13.814434 |

FIGURE 4.3: Extracting MixedLM random effects coefficients for data with 8 groups and 3 predictors

2. **Extracting Coefficients:** After fitting the data, random effects coefficients are extracted from the model summary. A dataframe containing intercepts and various slopes of fixed effects predictor features is prepared. See the figure 4.3

3. **Dimensionality Reduction:** Apply dimensionality reduction techniques to simplify the extracted dataframe and prepare the data for clustering. Based on the data characteristics, a suitable dimensionality reduction technique such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) is selected. See the figure 4.3

4. **Apply Cluserting (Evaluation):** The final phase involves the application of various clustering methods to the dimensionally reduced data. It involves running the clustering algorithm on a range of cluster numbers. Through this iterative process, an optimal number of clusters can be found in a large number of groups.

Optimal cluster number can be selected based on specific needs and after further analysis of the data. The performance is evaluated with the measure MSE (Mean squared error). Note that, on each clustering iteration the data is trained and tested on the model according to new cluster labels.

### 4.3.2 Shapley Value Explainer

**An alternative approach for extracting distinctive characteristics of groups involves using Shapley values.** Unlike the previous method that relies on the MixedLM model, which is suitable primarily for linear data relationships, Shapley Values can handle both linear and non-linear data. The Shapley values allow us to determine the contribution of each group within the dataset [26]. By analyzing these contributions, we can learn more about how different groups influence the overall patterns observed in the data. These contributions can be considered as unique attributes of groups.

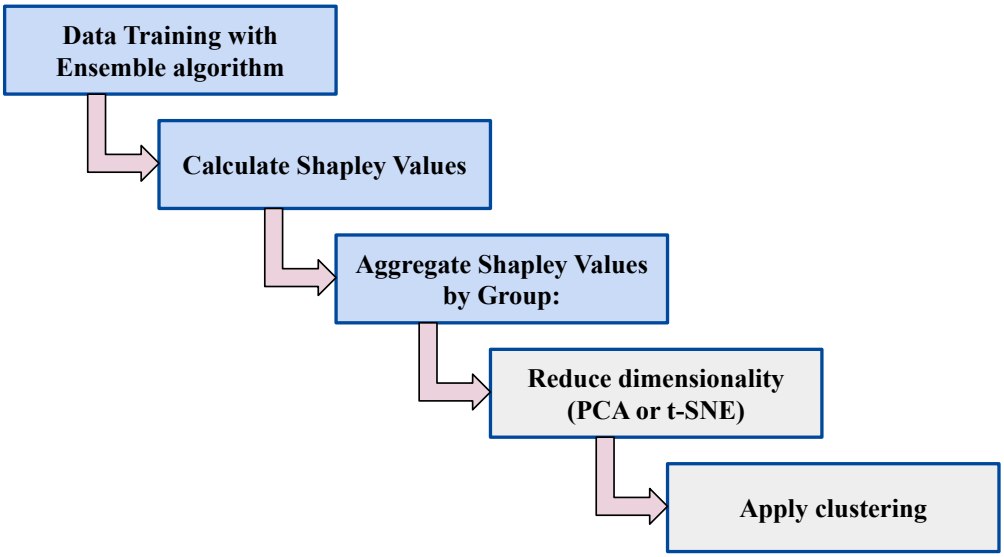



FIGURE 4.4: Method: Using Shapley Value Explainer

The method shown in figure-4.4 has been divided into 5 steps:

1. **Data Training with Ensemble Algorithm:**
   Similar to the previous method, the entire data is trained on an ensemble algorithm. In this study, the random forest model is considered.

2. **Calculate Shapley Values:**
   Compute the Shapley value of each data point using a Shap Explainer. This calculates Shapley values for each data point, including a value for each feature. A dataframe containing all these Shapley values is created.

3. **Aggregate Shapley Values by Group:**
   In order to find an explanation of a single group, we aggregate the Shapley

values for all data points belonging to the same group. This can be done by simply calculating the mean absolute sum of Shapley values within each group. Now, we have a measure of how much each feature including the grouping feature contributes.

4. **Dimensionality Reduction:**
   This step remains optional and depends on the number of features available in the data. If dealing with many features, dimensionality reduction techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) can be used. If the method uses Jenks Natural Breaks for clustering, which only works with one-dimensional data, then dimensionality must be reduced to one-dimensional data.

5. **Apply Clustering (Evaluation):**
   Here, the K-means clustering or Jenks Natural Breaks can be used for clustering. The clustering is used as a performance evaluation step. We experiment with varying the number of clusters and assess how the performance changes after clustering.

**For selecting the optimal cluster**, we have considered the Elbow method (Kneed locator), which provides the optimal cluster based on the performance results of the clustering. Although, visual inspection of a plot and further analysis are required as such a process can not be automated, this approach serves as a good starting point.

# Chapter 5

# Experimental Setting

In this section, we show the process of synthetic data generation employed in this study. Using the synthetic data, we want to conduct experiments on data with an increasing number of groups and evaluate our proposed methods. The generation of this synthetic data is executed in three primary steps: starting with creating data with fixed effects features. Next, we explain our grouping terminology and how the groups are created in the data. Lastly using those groups, we show how we incorporate random effects for various cases.

## 5.1 Synthetic Data Generation

### 5.1.1 Fixed Effects

Earlier in the section 1, we discussed that fixed effects have a consistent influence on the dependent feature. To generate the fixed effects component of a linear mixed-effects model, we start by simulating linear regression data with multiple features.

Consider the equation 2.1,

$$Y_{fixed} = X\beta + \epsilon$$

The $X$ is of dimension $N \times m$, values sampled from a uniform distribution $U(a, b)$, where $m$ is the number of predictive features. The coefficients ($\beta$) and global error term ($\epsilon$) follow a normal distribution $N(\mu, \sigma^2)$.

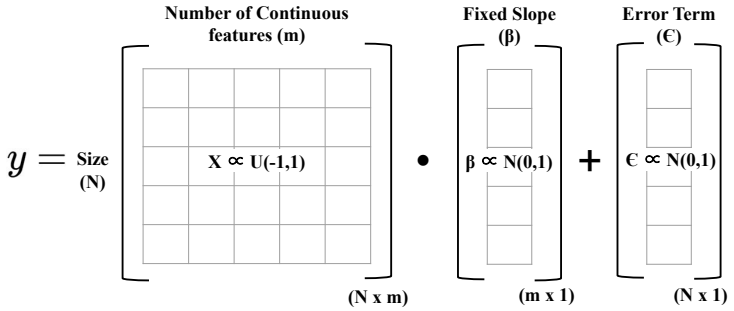



FIGURE 5.1: Fixed-effects Regression

In this study, we have considered fixed effects features in $X$ follows a $U(-1, 1)$ distribution with a dataset of size $N = 1000$, slopes ($\beta$) and error ($\epsilon$) sampled from a $N(0, 1)$.

### 5.1.2 Group Splitting

The groups are an essential feature to incorporate random effects in data. The random effects can be applied to specific groups or clusters. Once the data with fixed effects is simulated, the next step would be creating groups, using which later the random effects will be added to the data. The data will be split into groups, which contain smaller groups. Similarly, real-world data, where data is naturally clustered such as schools within states or countries, makes data structure more complex.

Here, we want to introduce the terminology which we used in this study with regard to groups i.e. Effective groups and Visible groups.

- **Effective groups**
  These are the primary, larger clusters created within the data. These groups may not be directly seen in the data but they are the ones which cause the random variation in the data. For example, in a country, the regions or states can be considered as effective groups.

- **Visible groups**
  These are the groups, which are created from the effective groups as if they are nested. They are mostly smaller and can be directly analysed. For example, in a region, schools and hospitals can be considered as visible groups.

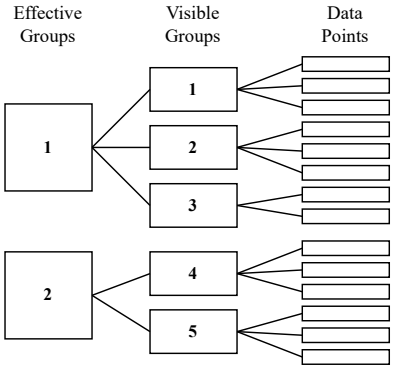



FIGURE 5.2: Nested Design

According to figure-5.2, the groups are formed in a way that smaller groups are nested within the bigger ones. Also, we assign nearly the same number of instances per group.

**Splitting.** The group splitting is in two steps. For example, the size of the data is 100. We want to form 3 effective groups and 7 visible groups.

**Step 1: Effective groups split**
The step simply divides the instances into three nearly equal parts. If there are any remaining instances, it will be added one by one from the first groups. By doing so we get {34,33,33}, consider figure 5.3

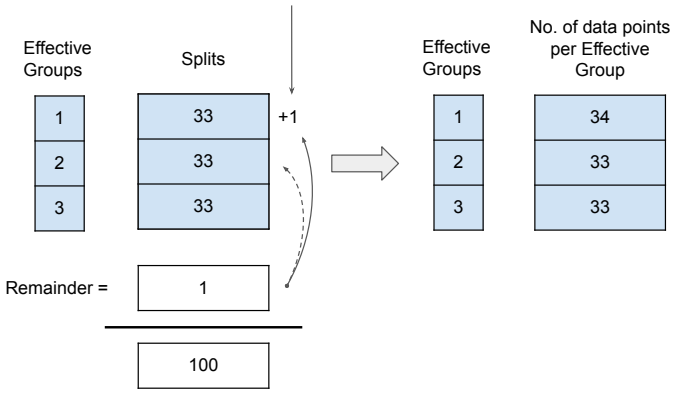



FIGURE 5.3: Effective Groups Split

**Step 2: Visible groups split**
We start by assigning how many smaller groups should be allotted to effective groups. Here it would be {3,2,2}. Then each effective group instance will be further split into corresponding allocations of smaller groups. For instance, the first effective group with 34 would be split into 3 smaller groups, consider figure 5.4

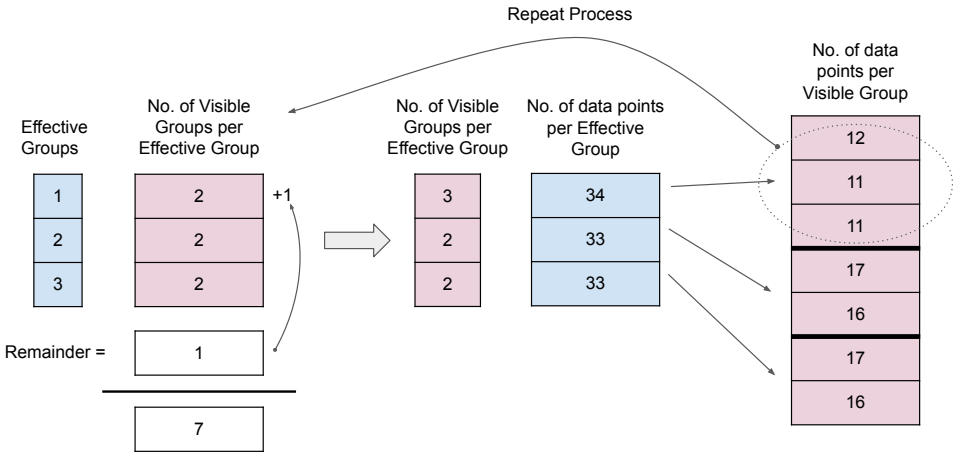



FIGURE 5.4: Visible Groups Split

Once the groups are created, they will be added to the dataframe along with the fixed effects features as categorical features. For method evaluation and experiments, we'll primarily focus on the visible groups, please see section 6.

### 5.1.3 Adding Random Effects

This section focuses on incorporating random effects using groups in data created with fixed effects. We show three cases to model random effects: intercepts only, slopes only, and both. For a detailed explanation of random effects and their types, please refer to section 2.1.

#### 5.1.3.1 Intercepts only ($u_{0j}$)

In this case, random intercept values $u_{0j}$ are added to the instances of each $j$-th group. This allows each group to have its own intercept and adds variability between groups. The modified equation of the random intercept model (see equation 2.3) can be represented below for 3 groups:

$$y = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix} + \begin{bmatrix} u_{01} + \varepsilon_{i1} \\ u_{02} + \epsilon_{i2} \\ u_{03} + \epsilon_{i3} \end{bmatrix} \tag{5.1}$$

Here, the $Y_{ij}$ is the data simulated for the fixed effects features (see section 5.1.1).

We use various ways to generate the intercepts $(u_{01}, u_{02}, u_{03})$ values. Please see section 5.2.1. The random effect error term $(\varepsilon_{ij})$ follows a $N(0,1)$, which is the error added to each group. The final target variable $y$ is the dependent variable formed with both fixed and random effects. The figure below shows the data simulated with mixed effects for the random intercept case.

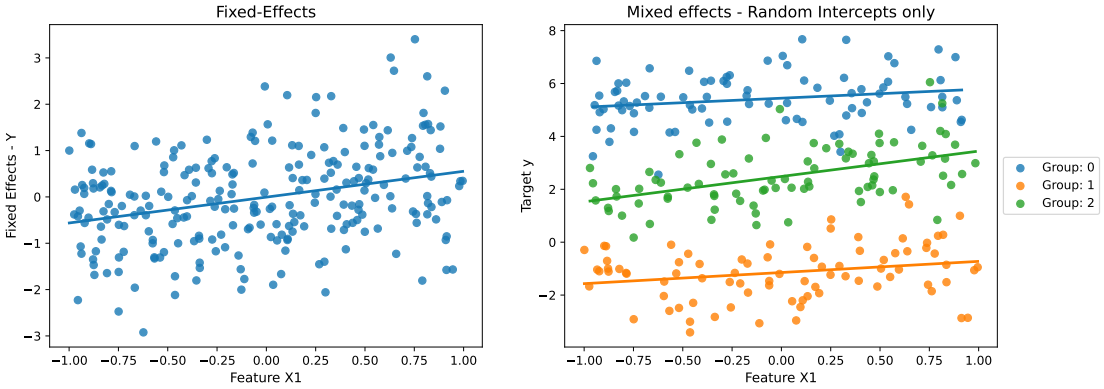



FIGURE 5.5: Synthetic data - Intercepts only: After creating fixed effects regression data, the random intercepts are incorporated and three groups have varying intercepts but the same slope.

#### 5.1.3.2 Slopes only $(u_{1j})$

The random slope values $u_{1j}$ interest with the fixed effects features. The interaction term $u_{1j} \cdot X_{ij}$ allows groups to have different slopes, adding complexity to the data. The modified equation of the random slope model (see equation 2.4) can be represented below for 3 groups:

$$y = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix} + \begin{bmatrix} (u_{11} \cdot X_{i1}) + \epsilon_{i1} \\ (u_{12} \cdot X_{i2}) + \epsilon_{i2} \\ (u_{13} \cdot X_{i3}) + \epsilon_{i3} \end{bmatrix} \tag{5.2}$$

Here, the $Yij$ is the data simulated for the fixed effects features (see section 5.1.1). The random slopes directly affect the fixed effects features, for the 3 fixed effects features, the above equation can be written as:

$$y = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix} + \begin{bmatrix} (u_{11} \cdot X_{i11} + u_{12} \cdot X_{i12} + u_{13} \cdot X_{i13}) + \varepsilon_{i1} \\ (u_{12} \cdot X_{i12} + u_{22} \cdot X_{i22} + u_{23} \cdot X_{i23}) + \varepsilon_{i2} \\ (u_{13} \cdot X_{i13} + u_{32} \cdot X_{i32} + u_{33} \cdot X_{i33}) + \varepsilon_{i3} \end{bmatrix} \tag{5.3}$$

We use various ways to generate the random slope values. Please see section 5.2.1. The random effect error term ($\varepsilon_{ij}$) follows a $N(0,1)$, which is the error added to each group. The final target variable $y$ is the dependent variable formed with both fixed and random effects. The figure below shows the data simulated with mixed effects for the random slope case.

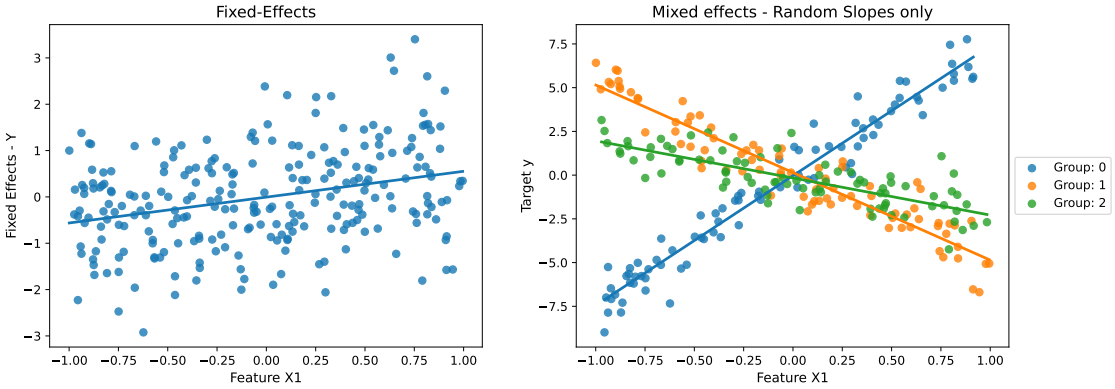



FIGURE 5.6: Synthetic data - Slopes only: After creating fixed effects regression data, the random slopes are incorporated and three groups have varying slopes but the same intercept.

#### 5.1.3.3 Intercepts and Slopes

This is the combination of the random intercept and slope case. It allows each group to have its own intercept and interacting slope values with fixed effects features. The modified equation of the random slope model (see equation 2.5) can be represented below for 3 groups and 3 fixed effects features:

$$y = \begin{bmatrix} Y_{i1} \\ Y_{i2} \\ Y_{i3} \end{bmatrix} + \begin{bmatrix} u_{01} + (u_{11} \cdot X_{i11} + u_{12} \cdot X_{i12} + u_{13} \cdot X_{i13}) + \varepsilon_{i1} \\ u_{02} + (u_{12} \cdot X_{i12} + u_{22} \cdot X_{i22} + u_{23} \cdot X_{i23}) + \varepsilon_{i2} \\ u_{02} + (u_{13} \cdot X_{i13} + u_{32} \cdot X_{i32} + u_{33} \cdot X_{i33}) + \varepsilon_{i2} \end{bmatrix} \tag{5.4}$$

Here, the $Yij$ is the data simulated for the fixed effects features (see section 5.1.1).

We use various ways to generate the random intercept and slope values. Please see section 5.2.1. The random effect error term ($\varepsilon_{ij}$) follows a $N(0,1)$, which is the error added to each group. The final target variable $y$ is the dependent variable formed with both fixed and random effects. The figure-5.7 shows the data simulated with mixed effects for this case.

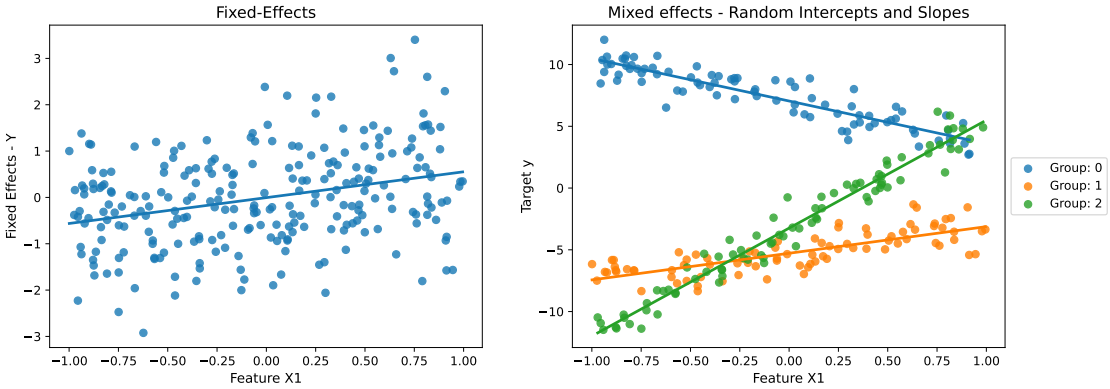



FIGURE 5.7: Synthetic data - Intercepts and Slopes: After creating fixed effects regression data, random intercepts and slopes are incorporated and three groups have varying intercepts with slopes.

## 5.2 Changeable Parameters

This section details the changeable parameters of the synthetic data generation function. These parameters help to control the characteristics of synthetic data, by adjusting the magnitude of random intercepts and slopes, the type of error introduced, and other factors. By adjusting these parameters, we have conducted experiments to assess model performance with an increasing number of groups.

### 5.2.1 Ways to Introduce Random Effects

The parameter controls the variation between either intercepts or slopes. The dataset should have as many random intercepts or slopes as the number of groups, hence how far or close the magnitude of intercepts or slopes should be, can be done by selecting one of the distributions.

We consider two ways to generate the values or magnitudes as shown in the figure-5.8, and 5.9.

1. **Linspace Distribution:** The values of intercepts and slopes are distributed evenly over a provided range, $(a, b)$. On top of that, we add the error of $N(0, 1)$ to make it random. See figure- 5.8,

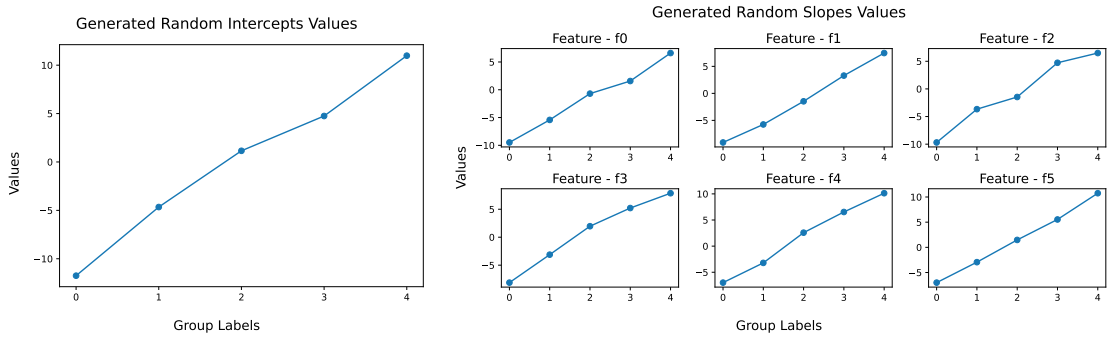



FIGURE 5.8: Random Effects with Linspace Distribution

with a range of (-10,10)

2. **Normal Distribution:** The values of intercepts and slopes are normally distributed over a provided mean and standard deviation value, $N(\mu, \sigma)$. This increases the complexity of the data, as the values are not evenly spaced but concentrate near a mean with some variation. See figure-5.9,

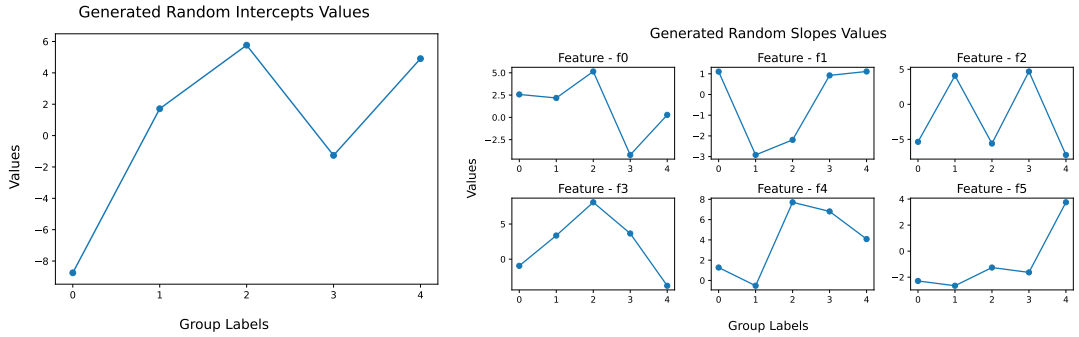



FIGURE 5.9: Random Effects with Normal Distribution

Normal Distribution of N(0,5)

In this study, we have mostly focused on random effects generated with linspace distribution. When evaluating the data with a large number of groups, this approach makes sure that no groups are getting overlapped, hence the performance of models can be properly evaluated.

### 5.2.2 Ways to Introduce Error Terms

This parameter helps to introduce errors into data. Using this parameter, data can be created with more complexity and the model's robustness can be evaluated. Here, we have used two approaches to add errors in the data.

1. **Error on each group:** This approach adds the error value to the data points of each group. It introduces additional variation in a group. The number of error values is the same as the number of visible groups in the data. Those values can be generated using the normal distribution of $N(\mu, \sigma)$.

2. **Error on entire target:** This increases overall noise in the data point. Errors are added directly to the target variable $y$, adding unexplained random noise to the data points.

### 5.2.3 Randomizing Labels

This parameter allows the group elements to shuffle. Initially, the grouping feature is labelled sequentially from 0 to $n$. By randomizing the labels, we can evaluate that the model does not rely on the specific order of group labels but rather learns the underlying patterns.

# Chapter 6

# Results

In this study, we are trying to determine the effect of the increasing number of groups on mixed-effects models and figuring out ways to mitigate any negative effects. To investigate this, we run several experiments. In Experiment-I 6.1, we show that there is a negative effect on the model performance as the groups increase. To solve this, in Experiment-II 6.2, we show the possible strategy to improve the performance of models by reducing the number of groups. Experiment-III 6.3 compares our methods to achieve the strategy, which shows the improvement in the performance of mixed-effects models on the same dataset configuration. Lastly, we apply methods to the real-world datasets.

For the following experiments, we use synthetic data. We have thoroughly described how we generate the synthetic in section 5. For conducting Experiments I and II, the parameters of the synthetic stay the same.

**Dataset Parameters:**

- Size of data = 1000
- Number of Predictive Feature = 5
- Way of adding Random Effects = Linspace Distribution, [-10, 10]
- Way of adding Error = Error on entire target (y), N(0,3)
- Shuffling Group Elements = Yes

## 6.1 Experiment-I: Increasing numbers of groups reduce the performance of mixed-effect models.

In this experiment, our objective is to examine the impact of an increasing number of groups on the performance of existing mixed-effects models. For that, we simulate the data with a fixed number of effective groups while progressively adding visible groups by maintaining consistency across datasets. The goal is to see any decline in the model performance if the visible groups keep on increasing without altering the effective groups, which is the original clustering of the dataset being evaluated.

We present results for the data with multiple predictive features and data inherits "Intercepts and Slopes" random effects. The additional data generation parameters for this experiment are given in the section-6.

- **Intercepts and Slopes:**

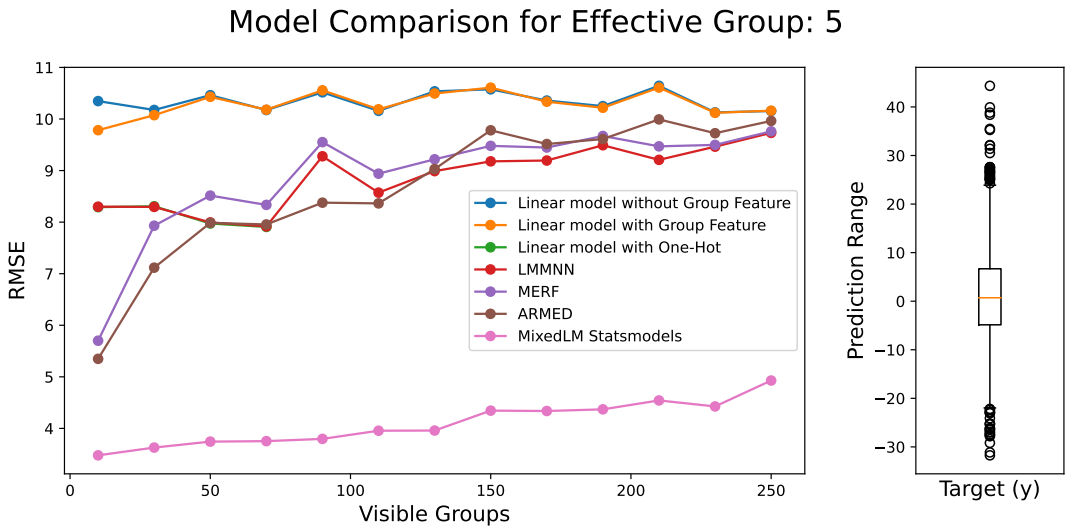



FIGURE 6.1: RMSE Performance comparison of models on the increasing number of visible groups. On the left is the prediction range for the Target(y) Variable for interpreting prediction quality.

The figure-6.1 is one of the scenarios of the entire experiment. The comparison of the various models is shown for the five effective groups in the data. The Root Mean Squared Error (RMSE) is shown here as a performance matrix, which allows comparison of the models on the same scale. As we can see, the performance of the models keeps on decreasing as the number of visible groups is

added to the dataset. Even the MixedLM model, which is best suitable for these kinds of datasets, shows an increase in error.

The entire results of the experiment are displayed in the figure-6.2. The experiment was conducted on various numbers of effective groups ranging from 3 to 8. The figure shows the individual performance of a model for varying effective groups, hence each model is shown on the individual performance scale.

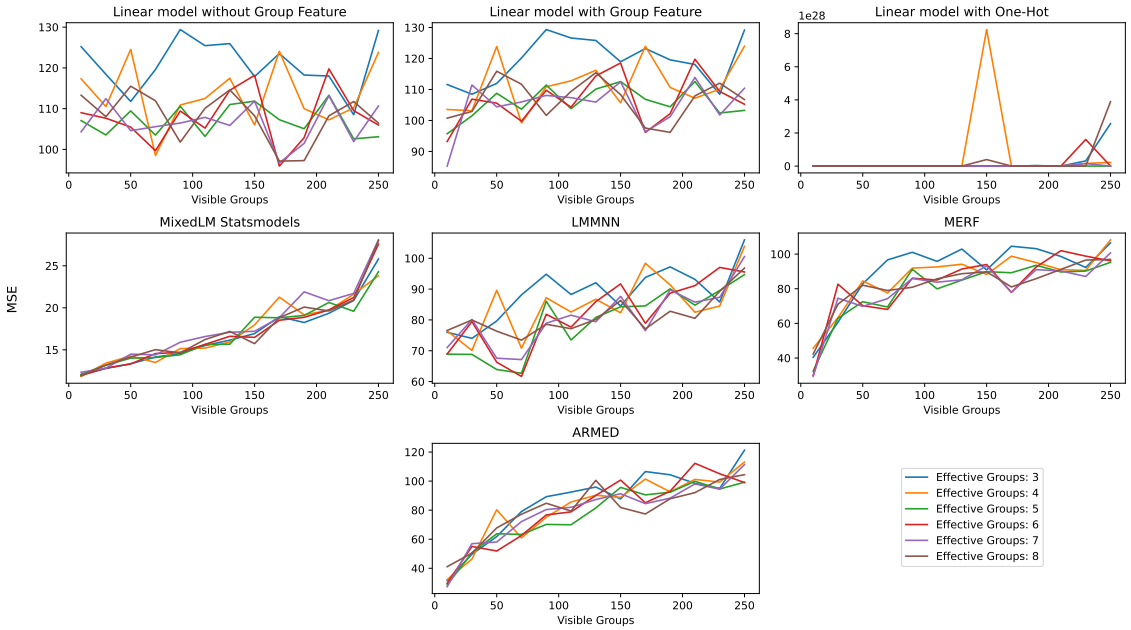



FIGURE 6.2: Experiment-I: Performance of models on data with an increasing number of visible groups for a number of effective groups (ranging from 3 to 8). **The scale of the performance matrix (MSE) is different for all models.**

The figure-6.2 showcases the effect of increasing the number of groups on various models. We can see that, firstly, not all models were able to perform better even on a lower number of visible groups. The Linear Model with One-Hot encoding showed performance which is not at all reliable for a larger number of groups. Secondly, the models which performed better on the lower number of visible groups, for example, MixedLM, eventually showed an increase in prediction error as the number of visible groups increased.

A similar kind of behaviour of the model has been seen on the same data with random effects type, Intercept only and Slopes only. The results for these random effects types can be found in the appendix section-8.

Overall from this experiment, we can conclude that the increasing number of

groups has shown a negative effect on the performance of the models. In the next experiment, we attempted to mitigate this effect by reducing the number of groups to improve model performance.

## 6.2 Experiment-II: Models consistently perform better with the original clustering of groups.

In Experiment-I, 6.1, we have seen the increasing number of groups reduces the performance of the model. To overcome this effect, we introduce a strategy to improve model performance. The strategy is to reduce the number of groups in the same dataset configuration and see if the prediction quality improves. In this experiment, we examine how models perform when the effective groups, which are the original clustering of visible groups, are used for evaluation, in contrast to Experiment-I, 6.1, where the visible groups were used as the grouping feature.

The data configuration stays the same for this experiment as given in section-6. The synthetic data provides the original clustering labels of the data as the effective group feature, and their subdivisions as visible groups feature. The model performance is evaluated individually, initially by only considering the effective groups feature and later with only the visible groups feature.

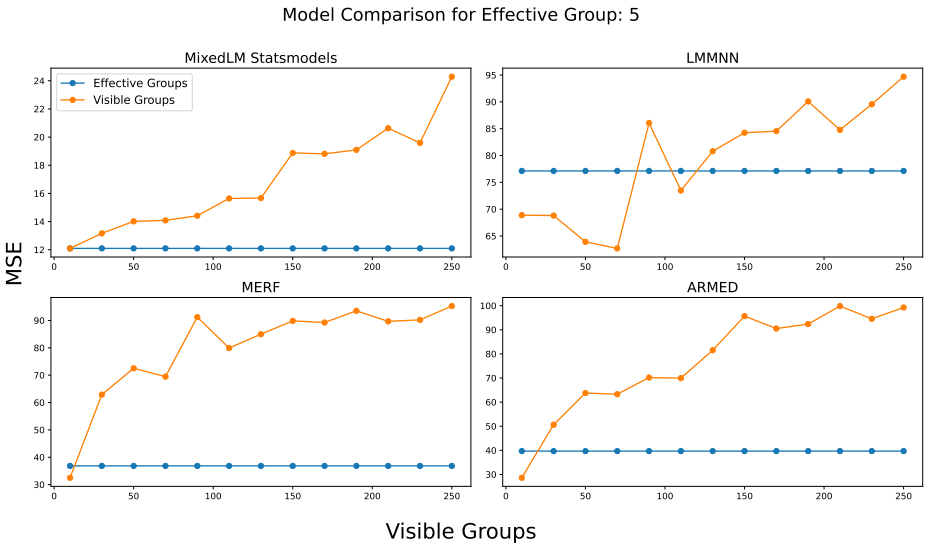



FIGURE 6.3: Comparison of the model performance evaluated on data with Effective Groups versus Visible Groups, Random effects type: Intercepts and Slopes. **The scale of the performance matrix (MSE) is different for all models.**

The figure-6.3 shows the performance of various mixed-effects models on data with five effective groups, on which the visible groups are increased from 10 to 250.

From the result, we can see the performance of models using effective groups improves the prediction quality. The improvement is observed for almost any number of visible groups. We can conclude that if the visible groups are somehow reduced, it can lead to better prediction quality. The models consistently show a similar behaviour for the data with other random effects types i.e., Intercepts only and Slopes only. For more detailed results, refer to appendix section-8.

Using this finding, we want to apply our proposed methods, which automatically reduce the number of visible groups by clustering them, thereby enhancing the model performance.

## 6.3 Experiment-III: Methods for reducing groups improve the prediction quality.

In the previous experiments, we have seen that increasing the groups affects the model performance negatively and reducing them through optimal clustering helps improve it. In this experiment, we apply our proposed method to follow the same strategy by automatically reducing the number of groups to show the methods do improve prediction quality. The detailed working of our proposed methods is discussed in the section-4, Methodology.

For applying methods, we use synthetic data. The applicability of methods varies with the number of groups and types of random effects. Therefore, a single experimental setup wouldn't be suitable to assess all the methods.

The results for our applied methods are as follows. For each method, the experimental setup is given as well.

**1. Heuristic Approach: Performance-based group removal**

The heuristic approach is applied with a function which aims to find an optimal clustering of groups in the form of subsets. The plot showcases how the method

keeps searching for a combination with better performance by iterating through possible combinations of groups.

We modified some of the synthetic data parameters given in section-6.

- Number of Visible groups = 10
- Number of Effective groups = 3
- Random-effects Type: Intercepts and Slopes
- Shuffling Group Elements = Yes

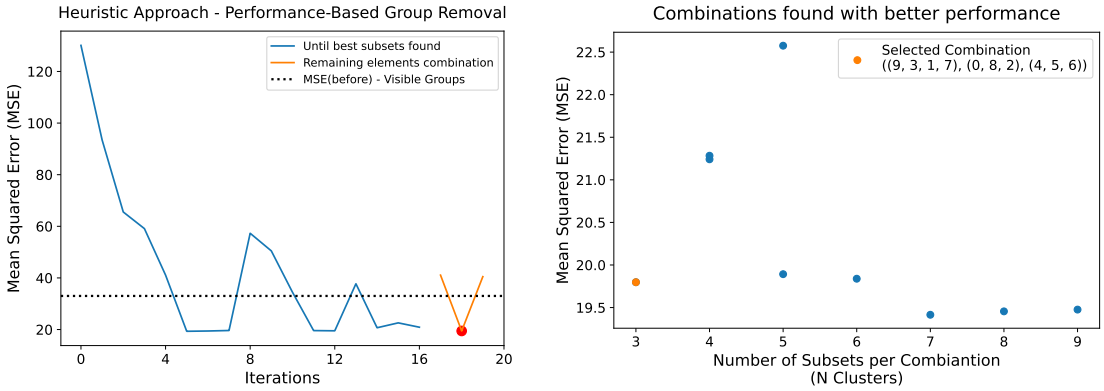



FIGURE 6.4: On the left: Performance matrix(MSE) varies through each iteration for the corresponding combination. On the right: The combinations, which performed better, are shown in terms of the number of subsets available in those combinations.

We can see in the figure-6.4, the method iteratively searches over better combinations which have lower errors than the threshold. The horizontal threshold line is nothing but the performance of the model using visible groups.

The method was able to find the clustering of groups which improves the performance of the model and it provided multiple combinations which does that.

**2. Heuristic Approach: Variance-based group removal**

A similar method works as a heuristic approach, which clusters groups based on variance, and also aims to cluster groups to allow a model to perform better. The result shows how the function evaluates different combinations and returns multiple ones with better prediction quality.

We modified some of the synthetic data parameters given in the section-6.

- Number of Visible groups = 10
- Number of Effective groups = 3
- Random-effects Type: Intercepts and Slopes
- Shuffling Group Elements = Yes

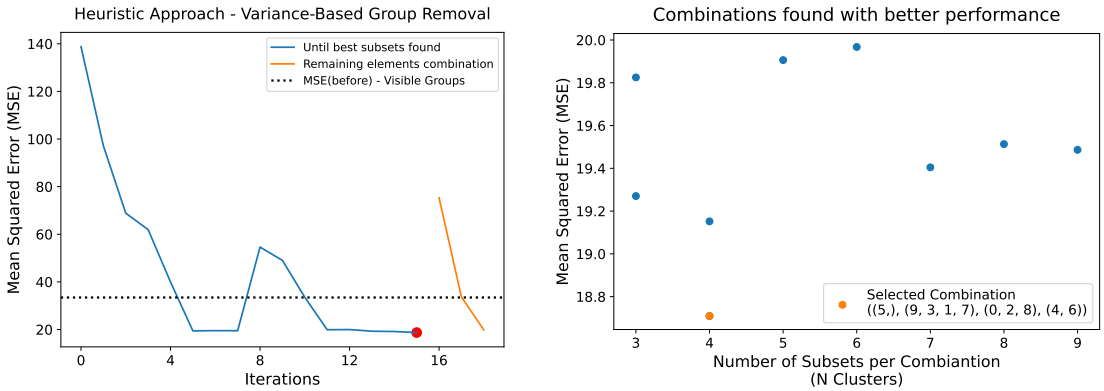



FIGURE 6.5: On the left: Performance matrix(MSE) shown for each iteration as the function searches through combinations. On the right: The combinations that performed better, are shown in terms of the number of subsets available in those combinations.

From the figure-6.5, we can see that the method was able to achieve better performance by forming subsets from groups. On the right plot, all the scatter dots denote a respective combination and all of them performed better.

Both two methods of heuristic approach Performance-based and Variance-based prioritizes finding a combination with the fewest clusters, even if there might be combinations with slightly better performance available.

**3. Unique Characteristics: Random Effects Coefficients**

The previous methods work with the mostly lower number of visible groups, this method works with a larger number of groups. The method extracts random-effects coefficients from the MixedLM model that somehow define each group in terms of unique measures. The information was then used to reduce the groups by clustering the group data.

We show that the method was able to improve the performance of the model. As the method works with a larger number of groups, we modified some of the synthetic data parameters given in section-6.

- Number of Visible groups = 200
- Number of Effective groups = 5
- Random-effects Type: Intercepts and Slopes
- Shuffling Group Elements = No

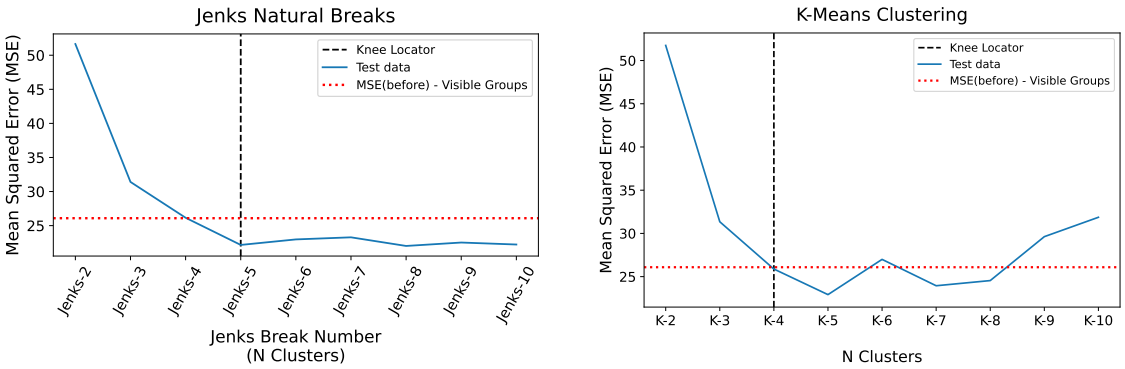



FIGURE 6.6: On the left: The clustering algorithm, Jenks Natural Breaks, is used to compare the performance of various numbers of clusters tried. On the right: The same plotted for K-Means clustering.

The figure-6.6 shows how the clustering algorithms tried various number clusters to find the one with better model performance. Here, the threshold is given, which represents the model performance on visible groups (here 200). The knee locator is used to choose the number of clusters after which the model performance does not change a lot.

The method was able to cluster groups using both clustering algorithms and the performance of the model is also improved. Next, we present our results for the method which does not rely on the MixedLM model and can independently do the same by using Shapley values.

**4. Unique Characteristics: Shapley Value Explainer**

In the previous method, we used the MixedLM model to get unique measures of groups. Here we consider the Shapley Value for getting the unique measure. The aim is to cluster the extracted values and put similar ones together as one.

Here, we show the result for this method that also improves the prediction quality of models. The method also can work with a larger number of groups but is restricted to the "Slopes only" kind of random effects. so we modified some of the synthetic data parameters given in section-6.

- Number of Visible groups = 200
- Number of Effective groups = 5
- Random-effects Type: Intercepts and Slopes
- Shuffling Group Elements = No

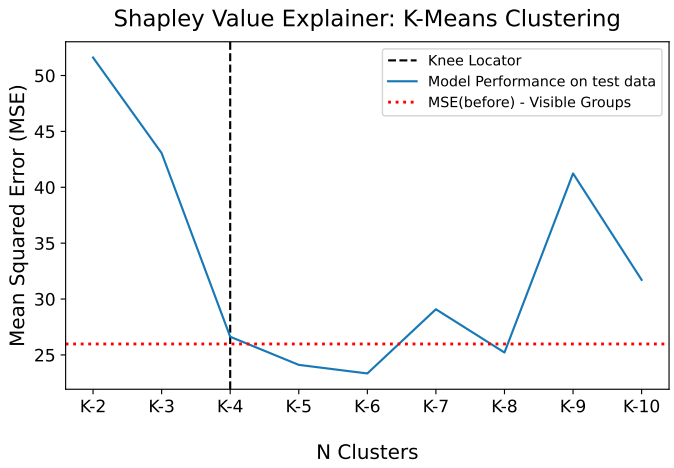



FIGURE 6.7: The K-Means clustering algorithm is used to compare the performance of various numbers of clusters tried and a knee locator to identify which number of clusters to choose.

In the figure-4.4, the horizontal red line acts as a threshold, it's the original performance, which we are aiming to improve. The knee locator chose three as optimal clustering but this might not always be true. We can see after three clusters, the model performance increases at a certain point and then becomes almost unchanged.

The method can work with a larger number of groups and also was able to improve the model performance. The method uses a clustering algorithm for searching an optimal number of clusters based on the performance recorded. The chosen N-clusters may not be the optimal one as several N-clusters surpass the performance threshold.

**Short conclusion.** With Experiment-III 6.3 on applying various methods to reduce groups, we have seen that methods follow the strategy from Experiment-6.2. Using those methods, the performance of the model can be improved. Even if the performance remains unchanged, the clustered groups may further improve interpretability.

## 6.4 Real World Datasets

The proposed methods in Experiment-III, 6.3, were able to reduce the groups in the data and showed an improvement in the prediction quality of the model. In this section, we apply our methods to real-world datasets. We present the results after applying a suitable method to the dataset whether the existing groups in a data can be clustered. We compare model performance by considering actual groups and after clustering them. The two existing mixed-effects datasets are considered.

### 6.4.1 The Sleep Study Data

The Sleep study data, the records of the average reaction time per day for subjects in a sleep deprivation study. The cognitive effects of varying sleep are tracked over multiple days and how lack of sleep affects the reaction time of the subject is recorded as reaction time (in milliseconds) [27], [28].

The method with random effects coefficient using MixedLM is applied. For clustering, the K-Means clustering and MixedLM model are used for evaluating the model performance on various clusters. For more details about the method used, please see section-4.3.

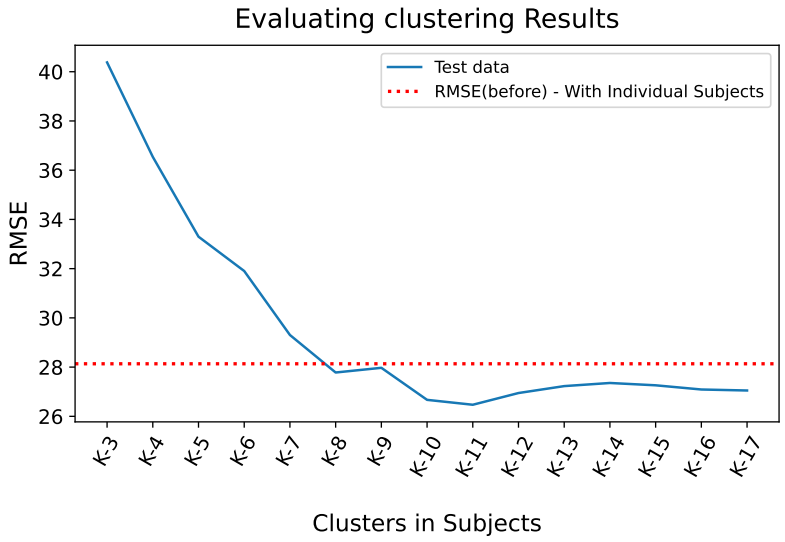



FIGURE 6.8: K-Means clustering applied on various K clusters. The test data performance versus its corresponding cluster number is plotted.

**Explanation.** The clustering algorithm evaluated clusters from 3 to 17 and found that performance improved after eight clusters ("K-8"). The table-6.1 shows for "K-8", how subjects are distributed across clusters.

| Subjects Clusters Found | Subject_ID |
|---|---|
| 0 | [334, 350] |
| 1 | [330, 331, 333, 352, 372] |
| 2 | [309, 310] |
| 3 | [337] |
| 4 | [332, 351, 369, 371] |
| 5 | [349, 370] |
| 6 | [308] |
| 7 | [335] |

TABLE 6.1: For, "K-8", the table details subjects' membership in each cluster.

Although, there are multiple solutions which provide better performance. In table-6.1, those clusters with more than one subject, suggest there is a similarity in the subjects and treating them as one cluster affects the performance positively.

### 6.4.2 The Dietox Data

The Dietox dataset, tracks the growth of 72 pigs under different dietary conditions. The weight of pigs is recorded over time, measured in weeks, after putting them on different diets [29], [30].

The method with random effects coefficient using MixedLM is applied. The aim is to find any similarities among pigs to group them and see if it affects the model performance for predicting their weight. For the method used, the K-Means clustering and MixedLM model are considered for evaluation. For more details about the method used, please see section-4.3.

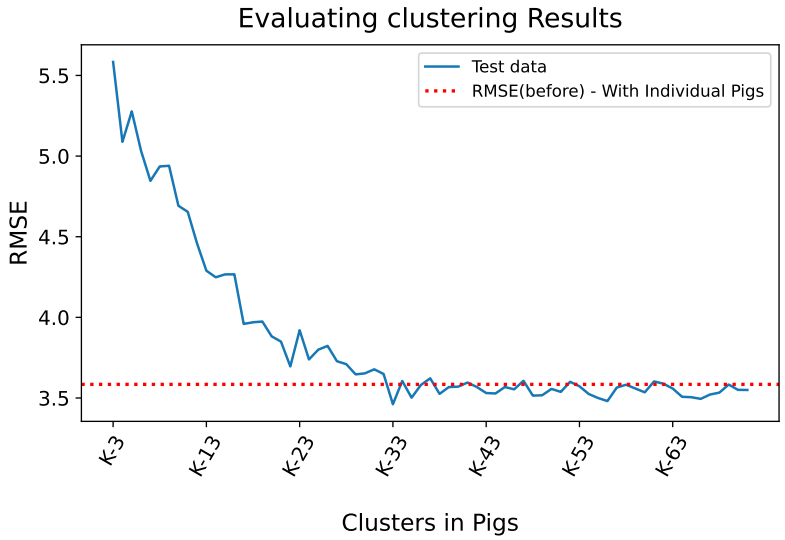



FIGURE 6.9: K-Means clustering applied on various K clusters. The test data performance versus its corresponding cluster number is plotted.

**Explanation.** The clustering algorithm evaluated clusters from 3 to 71 and found that performance improved right after 33 clusters ("K-33"). The table-**??** shows for "K-33", how the clusters are found among pigs.

| Pig Clusters Found | Pig_ID |
|---|---|
| 0 | [4603, 8191] |
| 1 | [4760] |
| 2 | [4817, 6912, 8442] |
| 3 | [6430] |
| 4 | [4757, 8193] |
| 5 | [4602, 4814, 6055, 8142] |
| 6 | [5497, 5851, 6287] |
| 7 | [4605, 4645, 4856, 5501, 8051] |
| 8 | [8144] |
| 9 | [4759, 5865] |
| 10 | [4756, 5500] |
| 11 | [6056] |
| 12 | [5524, 5850, 8141] |
| 13 | [4601, 5581, 6910, 8437] |
| 14 | [8195, 8269] |
| 15 | [6057, 6058, 6207, 6208, 6211, 8439] |
| 16 | [6432] |
| 17 | [4854, 4857] |

Continued on next page

| Pig Clusters Found | Pig_ID |
|---|---|
| 18 | [5392] |
| 19 | [5862, 5866] |
| 20 | [5578, 6433, 8050] |
| 21 | [4643, 4858, 5582] |
| 22 | [5502] |
| 23 | [5389, 8139] |
| 24 | [8192] |
| 25 | [4813] |
| 26 | [8273] |
| 27 | [4815, 5852, 6909, 8270] |
| 28 | [5528, 8271] |
| 29 | [6284, 6288, 8053] |
| 30 | [4641] |
| 31 | [5527] |
| 32 | [8049] |

TABLE 6.2: For "K-33", the table details the distribution of pigs across 33 clusters.

Similarly, there are multiple cluster values are found which also provide better performance than with considering individual pigs. In table-6.2, those clusters with more than one pig, suggest that there is a similarity in those pigs and treating them as one group, avoids redundancy and improves model performance.

**Summary.** One of the proposed methods is applied to two real-world datasets, aiming to enhance the performance of an existing model. The method evaluated every possible cluster within the groups. Specifically, for each cluster, the model had to be retrained with data. Overall, the method was successfully able to combine similar groups and showed improved model performance.

# Chapter 7

# Discussion

In this chapter, we thoroughly discuss the findings presented in the chapter-6, Results. We previously demonstrated the negative effect of an increasing number of groups on model performance and suggested a strategy to overcome this challenge using synthetic data. Firstly, we will discuss the results of Experiments I and II, particularly how the performance of various models deviates with an increasing number of groups and how our proposed strategy improves the performance, respectively. We will then compare the effectiveness of our methods employed for group reduction, and their limitations in different scenarios. Lastly, we will address the interpretability aspect of our methods.

## 7.1 Experiment-Specific Discussions

1. **Experiment-I**

   In Experiment-I, 6.1, we split effective groups into smaller visible groups ranging from 10 to 250. Existing mixed-effects models including simple linear models, are then used to evaluate the performance on the same data configuration. The primary metric for evaluating performance was MSE, and RMSE for model comparison. We demonstrated the results for the data with "Intercepts and slopes" random effects.

   In the fig-6.1, initially, with 10 visible groups, the MixedLM model exhibited the lowest RMSE(3.478), outperforming the other models. ARMED and MERF were also able to handle the groups, with RSMEs of 5.349 and 5.70, respectively. However, as the number of visible groups added up to 250,

MixedLM performance merely deviated with the increased RMSE(to 4.928). In contrast, ARMED and MERF significantly showed bad performance with RMSEs worsening to 9.963 and 9.76, respectively.

Regarding the fig-6.2, the trend was clear for all effective groups, as the number of visible groups increased, model performance declined. The results of the entire experiment are shown in fig-6.2, where the performance of each model is recorded with the individual scale of error (MSE). The linear model with one-hot encoding introduced significant bias beyond a certain number of visible groups and recorded a huge amount of error. Even though the comparison in the fig-6.1 only extends up to 190 groups, even with fewer groups the performance is unacceptable.

The LMMNN on the other side was not able to model this particular type of random-effects, as even with fewer groups, the RMSE(8.299) was high. The linear models also could not handle the data with random effects due to a lack of information about the group dynamics, suggesting that the grouping feature needs to be handled explicitly for modelling the data with random effects.

In summary, the mixed-effects model which performed well initially, showed an increase in error. Even the MixedLM model, which was the best fit for the linear mixed-effects data showed a slight shift in the error, indicating the higher number of groups does affect the prediction quality of the model.

2. **Experiment-II**

In the previous experiment, 6.1, we evaluated the performance of groups by considering visible groups for various mixed-effects models. In this experiment, we showed the same for considering the effective groups, the models performed well. We assumed that the effective groups are the feature that contains the original clustering of visible groups.

Utilizing the same dataset configuration as in the Experiment-I, 6.1, the experiment individually compares the MSE for models on different scales. Here, the MSE for effective groups remained stable because the dataset stayed the same for the particular effective groups (specifically, 5 in this case).

For the fig-6.3, among the models evaluated, the error rate of MixedLM may have increased for 250 visible groups (24.289) but when effective groups

for modelling are considered, there is a significant amount of drop in MSE (12.096). For MERF as well, the MSE for effective groups was 36.83, which was improved from 95.255. ARMED on the other hand showed a similar performance as MERF and the MSE dropped from 99.256 to 39.688 upon considering effective groups for modeling.

As the LMMNN is not able to model random effects with both "Intercepts and Slopes", yielded unreliable MSE results. However, the data with "Intercept only" random effects, LMMNN shows a similar trend and improvement as other models.

In summary, this experiment validates our proposed strategy to improve the performance of models. The results from this experiment clearly show that clustering groups to reduce them helps reduce the error rates.

3. **Execution time of mixed-effects models in experiments**

In these experiments, the models had to run multiple times to evaluate the data. The run-time of models for a number of visible groups is given in the fig-7.1.

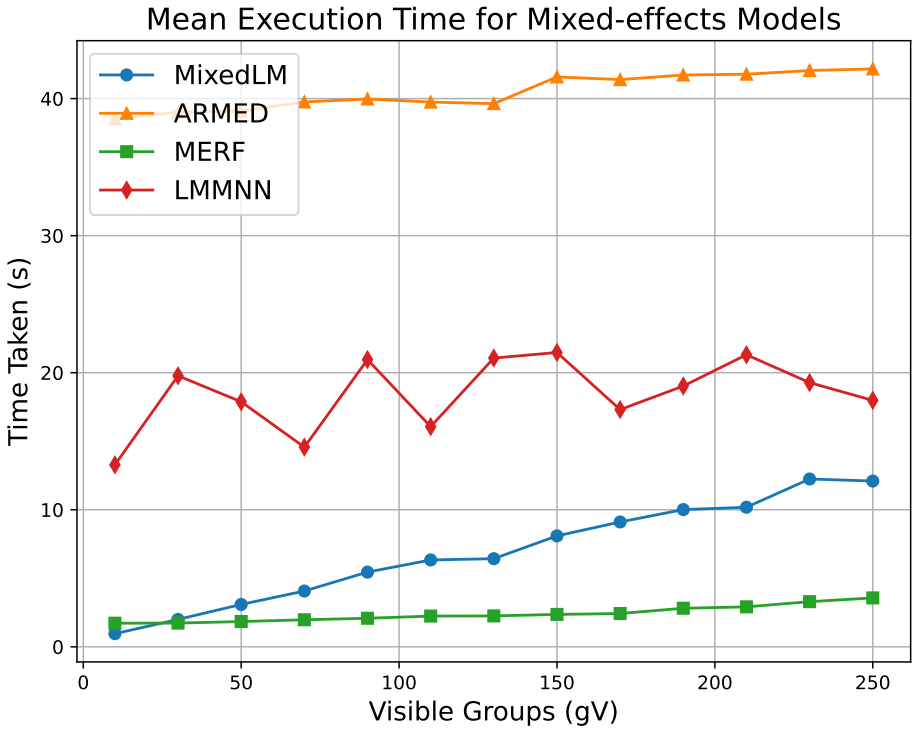



FIGURE 7.1: Comparison of the execution time for models with respect to a number of visible groups.

The deep learning and tree-based models are considered in this study for evaluation along with Python's MixedLM model. In the fig-7.1, the MixedLM model run-time is also increased with the number of visible groups. Hence, the reduction in the group also results in an improvement in the model's run-time.

## 7.2 Methods Comparison and Limitations

The section discusses the results of Experiment-III, 6.3, where we applied our proposed methods for group reduction. The aim is to understand to what extent the methods were able to reduce groups to improve the performance of models and provide limitations based on whether they achieved optimal clustering.

1. **Experiment-III**

   **Regarding the heuristic approach,** both methods, Performance-based and Variance-based group removal, were able to find combinations with lower MSE. The data configuration was the same for both methods and contains 10 visible groups. For both methods, the initial error rate for visible groups was around $\tilde{3}3$, which methods are aiming to improve.

   In the fig-6.4 and 6.5 (left plots), the algorithm searches through the combinations in two stages. The first stage involved identifying well-performed subsets of groups from a set of visible groups illustrated through a blue line plot. This step focused on groups that naturally show better performance when put together. In the second stage, the algorithm tried reintegrating group elements excluded from these subsets through all possible combinations, checking whether they further improved performance. The algorithm always picks the combination with the lowest number of subsets and has the lowest error rate (MSE).

   The Performance-based group removal method showed a reduction in MSE to 19.426 and found the combination with three subsets, - ((9, 3, 1, 7), (0, 8, 2), (4, 5, 6)). Similarly, the Variance-based method provided a comparatively better result, bringing the MSE down to 18.709 and found the combination with 4 subsets, $((5,), (9,3,1,7), (0,2,8), (4,6))$. Apart from these, the methods found multiple combinations with better performance.

**For Unique Characteristics Methods,** fig-6.6 and 6.7, clustering algorithms for evaluating the MSE on test data is used. There are two ways to choose an optimal number of clusters, as after a certain number of clusters the MSE does not change drastically. 1) Kneed Locator, 2) Visual Inspection. Here, we have used the Kneed Locator to pick the optimal number of clusters, but that will not always provide the optimal clusters in data hence visual inspection is also required.

The method with Random Effects Coefficients, refer to fig-6.6, showed a reduction in MSE for both clustering algorithms. Initially, the MSE was 26.088 for data with 200 visible groups. The Jenks Natural Breaks clustering, after 4 Breaks recorded a reduction in MSE for the rest of the Breaks value. The kneed locator picked "Jenks-5", which is 5 clusters, where MSE reduced to 22.997. Similarly, for K-Means clustering the kneed locator picked the "K-4", which is 4 clusters, where MSE was 25.879. However, if we visually inspect the error rate, "K-5" indicated an even better performance with MSE (22.918). For the rest of the cluster values the error rate again increased, suggesting 5 clusters as the optimal choice.

The last method with Shapley Value Explainer, refer to fig-6.7, applied on the same data setting for data with 200 groups. Here, the kneed locator picks "K-4" as optimal clusters. At "K-4" the test data performance was 26.626, which is slightly higher than the threshold value (26.088). The visual inspection clearly indicated that "K-6" should be chosen as it has the lowest MSE (23.341), after which performance was unstable and reached higher than the threshold value.

**Real-world dataset.** The random effects coefficients method was also applied to real-world datasets. For both datasets, method found the multiple solutions. On the Sleep Study data with 18 subjects, fig-6.8, at "K-8" the RMSE dropped down from 28.135 to 27.780, suggesting combining subjects into 8 clusters. Similarly, for the Dietox data with 72 pigs, fig-6.9, the best performance was found with "K-33", where the RMSE was improved from 3.584 to 3.461.

### Method Applicability

The methods were checked on the synthetic dataset size of 1000 instances and visible groups were increased from 10 to 250. The methods' applicability is compared with various group sizes, see table-7.1.

| **Methods** | **Number of Groups** |
|---|---|
| Brute Force Approach | Up to 20 |
| Heuristic: Variance Based | Up to 20 |
| Heuristic: Performance Based | Up to 50 |
| Random Effects Coefficients | Up to 250 |
| Shapley Value Explainer | Up to 250 |

TABLE 7.1: Methods Comparison on Group Sizes

The table-7.1 details the best possible application scenarios for each method. The categorization allows choosing the most appropriate method for data with a specific amount of groups. It is important to note that the actual performance can vary based on how the data is simulated and methods may work well in other scenarios as well.

### Execution Time of Methods

After stating the method applicability for each method, it is important to compare the runtime of these methods. We have considered the faster linear mixed-effects models for the application of methods i.e., MERF. Refer to fig-7.1 for the average execution time of the models used.

| **Methods** | **Number of Groups** | | | | |
|---|---|---|---|---|---|
| | **10** | **20** | **50** | **100** | **250** |
| Brute Force Approach | >3600 min | *NA | *NA | *NA | *NA |
| Heuristic: Variance-Based | <0.6 min | <2 min | <15 min | *NA | *NA |
| Heuristic: Performance-Based | <3 min | <40 min | *NA | *NA | *NA |
| Random Effects Coefficients | **<1 min | **<1 min | **<1 min | **<1 min | **<1 min |
| Shapley Value Explainer | **<1 min | **<1 min | **<1 min | **<1 min | **<1 min |

TABLE 7.2: Execution Time of Methods

*NA = Not Applicable, mostly when a method cannot handle many groups or becomes impractical due to exponentially increasing execution times.
**<1 min = Here, the methods use a predefined range of cluster sizes and the computational time depends on which model is used for evaluation. For slower models like ARMED and LMMNN, it may take longer.

In the table-7.2, the performance-based heuristic method worked comparatively slower than the variance-based method, as the model has to evaluate all the combinations first and then decide which group element to exclude. Beyond 20 groups, these methods become inapplicable. Similarly, the random effects coefficients and shapley value explainer methods have used 9 cluster sizes ranging from 2 to 10, meaning the model ran 9 times. For larger datasets with a higher number of groups, the model needs to be evaluated on more clusters, eventually requiring more time. Every evaluation of a combination or cluster model needs to be retrained, which can be crucial for deep-learning models.

2. **Limitations**

The developed methods applied to datasets with very constrained settings. In order to evaluate methods, a sufficient amount of error and random effects were incorporated to have data replicate the real-world scenario. Our key assumption for group reduction methods was that the visible groups already inherit an optimum number of clustering, which are methods aimed

at finding out. However, the methods were able to improve the performance but lacked capturing the optimal solution. In addition, some methods contain hyperparameters that need to be set before applying, which may restrict the methods to efficiently.

**Heuristic Approaches.** The heuristic approaches can provide multiple solutions, and the algorithm selects the approximate solution, but it does not provide an optimal solution every time. Consider the methods results from the figure-6.4 and 6.5,

| Combination | MSE | Number of subsets |
|---|---|---|
| ((4, 5, 6), (9, 3, 1, 7), (0, 2, 8)) | 19.276 | 3 |
| **((5), (9, 3, 1, 7), (0, 2, 8), (4, 6))** | **18.719** | **4** |
| ((0, 2, 8), (6), (4), (5), (9, 3, 1, 7)) | 19.903 | 5 |
| ((0, 2), (8,), (6,), (4,), (5,), (9, 3, 1, 7)) | 19.965 | 6 |
| ((9, 3, 1, 7), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.398 | 7 |
| ((9, 3, 1), (7,), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.565 | 8 |
| ((3, 1), (9,), (7,), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.516 | 9 |

TABLE 7.3: Variance-based group removal, multiple combinations with better performance than MSE with visible groups (33.38).

In the table-7.3, the selected combination by algorithm is highlighted. However, it was not the optimal clustering of groups, and on top of that, the algorithm found multiple solutions, which requires additional analysis to find the optimal solution, here it was ((4, 5, 6), (9, 3, 1, 7), (0, 2, 8)).

| Combination | MSE | Number of subsets |
|---|---|---|
| **((9, 3, 1, 7), (0, 8, 2), (4, 5, 6))** | **19.446** | **3** |
| ((4, 5, 6), (2,), (9, 3, 1, 7), (0, 8)) | 20.715 | 4 |
| ((0, 2, 8), (6,), (4,), (5,), (9, 3, 1, 7)) | 19.670 | 5 |
| ((4, 6), (2,), (5,), (9, 3, 1, 7), (0, 8)) | 22.605 | 5 |
| ((0, 8), (2,), (6,), (4,), (5,), (9, 3, 1, 7)) | 19.526 | 6 |
| ((9, 3, 1, 7), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.336 | 7 |
| ((9, 3, 1), (7,), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.390 | 8 |
| ((9, 3), (1,), (7,), (8,), (0,), (2,), (6,), (4,), (5,)) | 19.584 | 9 |

TABLE 7.4: Performance-based group removal, multiple combinations with better performance than MSE with visible groups (32.78).

Similarly in the table-7.4, the algorithm actually finds the optimal solution, but there are other solutions available which have even better performance.

**Time per iteration.** The Variance-based method is quicker because it runs the model just once each time to decide on a specific group to exclude. On

the other hand, the Performance-based method comparatively takes more time because it runs the model as many times as there are groups to figure out which one to remove.

Overall, our heuristic methods do not work with larger groups and require a lot of time without a guaranteed solution, see the table-7.2.

**Regarding methods with Unique Characteristics,** the clustering algorithms went through a range of cluster numbers set by the user. For each possible number of clusters, the algorithm records MSE on test data. These hyperparameters need to be set with an analysis because when dealing with larger groups, the execution time will also increase depending on the model used for evaluation.

The method with Shapley Values utilises the Tree Explainer to provide the local explanation of each data point. If the method is applied to a bigger dataset with high dimensionality and contains a larger number of groups, then the execution time may increase drastically. Similarly, with the random effects coefficients method with the MixedLM model, for a bigger dataset, it will require a lot of time to get random effects coefficients.

## 7.3 Interpretability

In this section, we provide the interpretability aspect of our proposed methods. We aimed to develop methods for group reduction, which not only enhance the performance of existing mixed-effects models but also provide an explanation of how the groups are combined to form a cluster. While Experiment-III, 6.3, demonstrated improved predictive performance of the models, the methods did not explain how groups are combined as anticipated.

**Heuristic Approach.** The heuristic approach was considered to overcome the computational limitations of the brute force approach, which also fails to provide an interpretable solution. The method of performance-based group removal lacks an explanation for why the groups within subsets were combined. It greedily searches for the combination that has better performance. The variance-based group removal method makes an informed decision for which group to exclude for forming a subset (a small cluster). It uses shapley values of groups for calculating variance, hence only information available is the variance structure. The

method is unable to provide the background information of the dataset features to understand on what basis the groups were combined.

The approach can be further improved by a strategy. The algorithm should utilise shapley values of groups and background information of existing features for selecting groups to guide the selection process for clustering. This way approach can make more rational decisions and can later provide interpretable results.

**Using unique characteristics.** The random-effects coefficients method utilises the ability of the MixedLM model to extract random intercepts and slopes for each feature of the dataset. The method then combines groups with similar intercept and slope values. However, this method does not clarify the significance of specific features of data within these clusters. The other method with the shapley value explainer, also follows a similar working principle for combining groups. The method combines groups based on feature importance and local data point explanations, which act as a unique measure of each group. It also fails to explain why those groups formed a cluster and what role a feature played in the clustering process. The reason can be understood from the example.

For example, consider a dataset of students in a country that contains district_id, school_type and student_id as features and is initially grouped based on student_id. In order to combine groups, the methods will not utilise the important features i.e., school_type and district_id, explicitly for clustering, even if it could be used as valuable background information. It shows a gap in the methods' ability to leverage all available data characteristics for group formation.

**Even on real-world datasets,** there were multiple clusters found and the best ones were chosen with visual inspection. In Sleep study data, why 18 subjects are clustered into 8 groups, is still unexplained. Also with the Dietox data, 72 pigs clustered into 33 groups, the reason behind the clustering is unknown. Hence, the methods fall short of explaining the reasons for selecting particular clusters or the similarities within combined groups, thereby limiting the interpretable results.

# Chapter 8

# Conclusion

The chapter summarizes the results and main findings of this work and lists perspectives for future work.

Our primary objective was to address challenges posed by a large number of groups in data with random effects. In Experiment-I, we found the existing mixed-effects models struggled to provide good performance in the large number of groups. We tried solving this issue by conducting Experiment-II, where the models' prediction quality improved when groups were combined using original clustering. We assumed that the groups in data follow an optimal clustering, using which it can be combined. To apply this finding, we developed methods for group reduction. Lastly in Experiment-III, our proposed methods successfully improved the performance of existing mixed-effects models. The appropriate method was also successfully applied to real-world datasets and was able to combine groups for better model performance. While the proposed methods only increase the predictive performance of models, they introduce a trade-off by not delivering significant improvements in interpretability.

**Future Work**

There are several areas that could be explored for further research based on the topics covered in this thesis. Here are some possibilities:

1. **Developing interpretable methods:** Our methods worked mostly to enhance the model performance and highlight a gap in interpretability, especially in understanding the rationale behind group clustering. Develop methods that primarily work on the interpretability aspect.

2. **Refineing synthetic data generation for complex scenarios:** In this thesis, we covered the basic type of mixed-effects data, focusing on only one grouping feature. However, real-world datasets often contain multiple categorical features, incorporating more complex group interactions. Developing such synthetic data would ensure the applicability of methods to diverse practical applications.

3. **Exploring non-linear feature relationships:** The thesis focused on the dataset exhibiting linear relationships. Future research could investigate the data with complex non-linear and mixed relationships. Optimizing model performance with such data can help solve the border real-world problems.

4. **Handling high-dimensional and large-scale data:** As mentioned in the thesis, we experimented with our methods with small-scale datasets. Future research should focus on large datasets characterized by high dimensionality, ensuring the methods maintain their relevance and applicability.

5. **Extension to classification tasks:** This study primarily concentrated on regression setting for developing methods. The work should be extended to a classification setting and utilize methods for classification settings to enhance model performance and interpretability.

These are just a few possibilities for further research based on the topics covered in this thesis. There are always new challenges to be addressed and opportunities to be explored.

# Appendix

## Experiments-I: Additional Results

The section covers additional results from Experiment-I, 6.1, where the impact of an increasing number of visible groups on the performance of mixed-effects models is investigated. The models recorded a decline in prediction quality for the other two types of random effects, i.e., Intercepts only and Slopes only.

1. **Intercepts only**

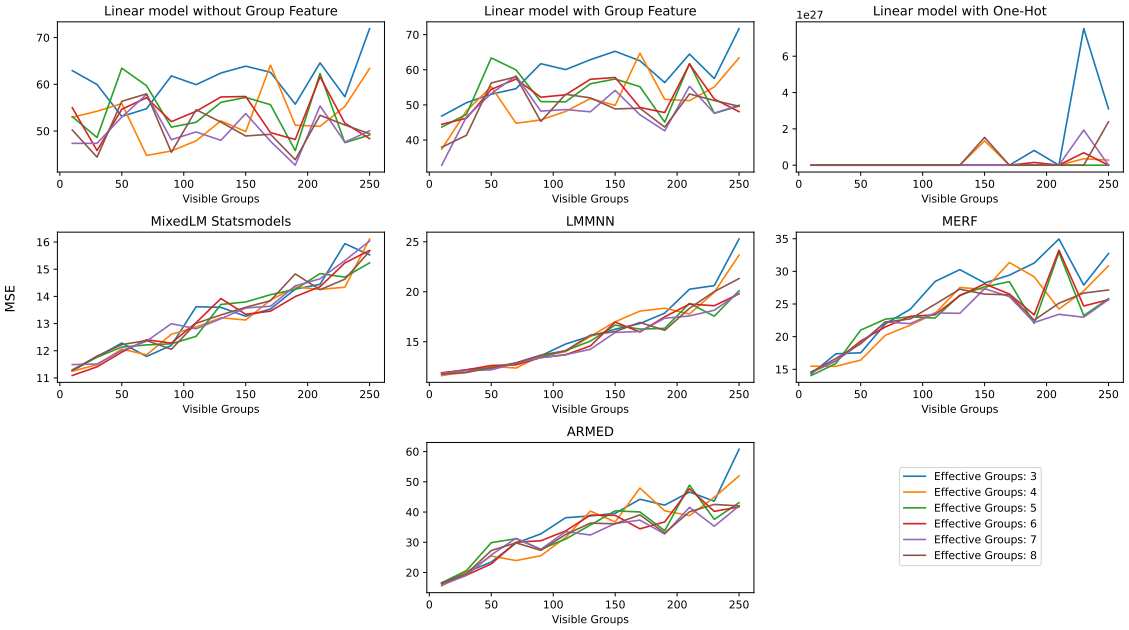



FIGURE 1: Performance of models on data with an increasing number of visible groups for a number of effective groups (ranging from 3 to 8). **The scale of the performance matrix (MSE) is different for all models.**

2. **Slopes only**

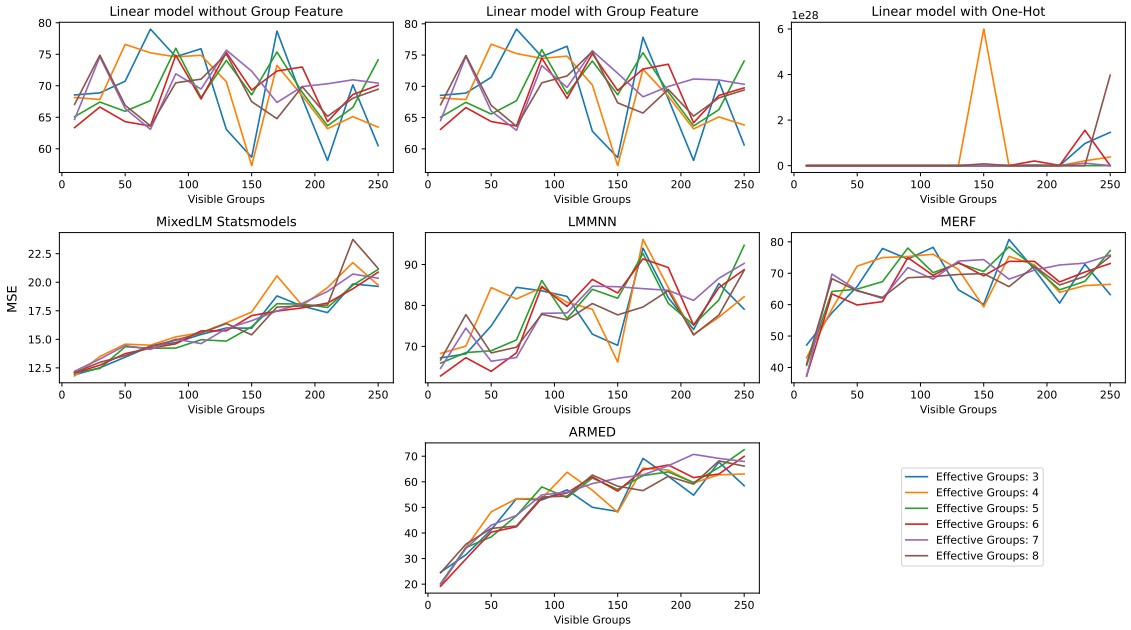



FIGURE 2: Performance of models on data with an increasing number of visible groups for a number of effective groups (ranging from 3 to 8). **The scale of the performance matrix (MSE) is different for all models.**

**Summary.** In the fig-1 and 2, the visible groups were increased from 10 to 250 and the prediction error was increased as it reached 250. This finding suggests that for all kinds of random effects, we considered in this study, the performance of the models was affected negatively.

# Experiments-II: Additional Results

This experiment aimed to reduce the number of groups in the same data configuration and examine whether the prediction quality of models improved, refer to section-6.2. It was assumed that small visible groups could be optimally clustered, and combining similar groups would affect the model performance. In our experimental setting, the optimal clustering labels are available as effective groups. The experiment compares the performance of the models with effective groups and with visible groups on the same datasets.

In this section, we provide the results for the other two kinds of random effects i.e., Intercepts only and Slopes only.

1. **Intercepts only**

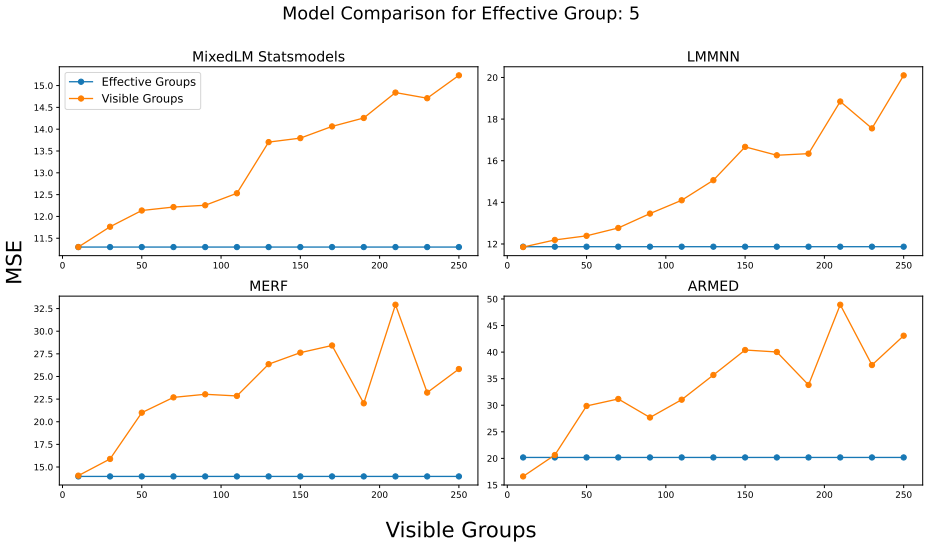



FIGURE 3: Comparison of the model performance evaluated on data with Effective Groups versus Visible Groups, Random effects type: Intercepts only. **The scale of the performance matrix (MSE) is different for all models.**

2. **Slopes only**

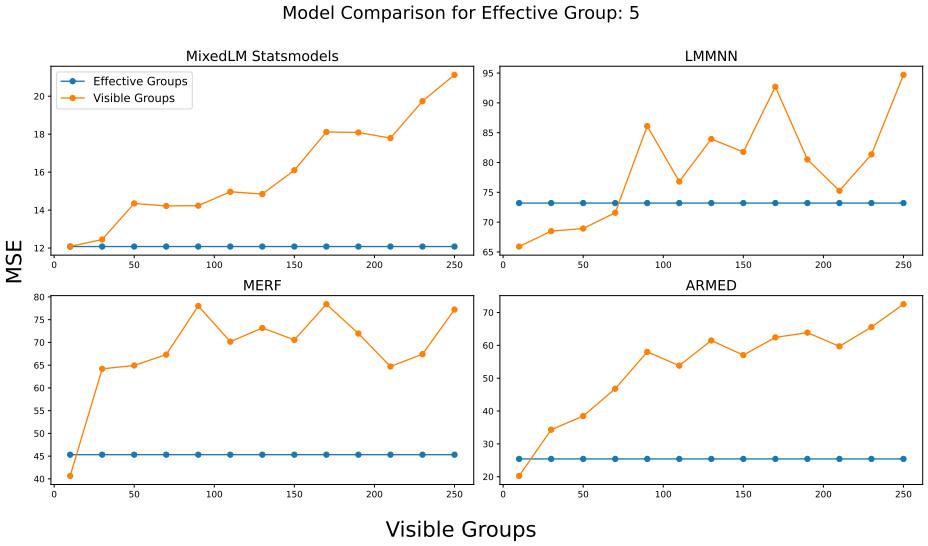



FIGURE 4: Comparison of the model performance evaluated on data with Effective Groups versus Visible Groups, Random effects type: Slopes only. **The scale of the performance matrix (MSE) is different for all models.**

**Summary.** In the fig-3 and 4, when the effective groups were considered over visible groups, the error dropped drastically. It suggests even for 'Intercepts only"

and "Slopes only" types of random effects, when similar groups are combined, models' prediction quality is enhanced. The models were able to better fit the grouping relationships with combined groups and we have applied the same finding for developing the methods that automatically reduce the visible groups in data.

# Bibliography


[1] Anthony S. Bryk Stephen W. Raudenbush. *Hierarchical linear models: applications and data analysis methods (2. ed., [3. Dr.] ed.).* Thousand Oaks, CA [u.a.]: Sage Publications, 2002.

[2] Dylan G.E. Gomes. *Should I use fixed effects or random effects when I have fewer than five levels of a grouping factor in a mixed-effects model?* PeerJ, 2022.

[3] Santawisook P. Wu Z. Chen T. *A multi-level model for analyzing whole genome sequencing family data with longitudinal traits.* BMC Proc 8 (Suppl 1), S86 (2014).

[4] Shenyang Guo. *Analyzing grouped data with hierarchical linear modeling, Children and Youth Services Review, Volume 27, Issue 6*. 2005. URL: https://doi.org/10.1016/j.childyouth.2004.11.017.

[5] Rauber A. Merkl D. Dittenbach M. *Uncovering hierarchical structure in data using the growing hierarchical self-organizing map*. Neurocomputing, 48, 199-216, 2002. URL: https://doi.org/10.1016/S0925-2312(01)00655-5.

[6] Yihui Luan and Hongzhe Li. *Clustering of time-course gene expression data using a mixed-effects model with B-splines*. Vol. 19. 4. Mar. 2003, pp. 474–482. URL: https://doi.org/10.1093/bioinformatics/btg014.

[7] Daniah Trabzuni, the United Kingdom Brain Expression Consortium (UKBEC), and Peter C. Thomson. *Analysis of gene expression data using a linear mixed model/finite mixture model approach: application to regional differences in the human brain*. Vol. 30. 11. Feb. 2014, pp. 1555–1561. URL: https://doi.org/10.1093/bioinformatics/btu088.

[8] Liang Ye Ye Shangyuan and Zhang Bo. *Bayesian Functional Mixed-effects Models with Grouped Smoothness for Analyzing Time-course Gene Expression Data*. Current Bioinformatics 2021; 16 (1), 2021. URL: https://dx.doi.org/10.2174/1574893615999200520082636.

[9] Dylan M. Nielson and Per B. Sederberg. *MELD: Mixed Effects for Large Datasets*. Cold Spring Harbor Laboratory, 2017. DOI: 10.1101/156315. URL: https://www.biorxiv.org/content/early/2017/06/27/156315.

[10] Fabio Sigrist. *A Comparison of Machine Learning Methods for Data with High-Cardinality Categorical Variables*. Vol. abs/2307.02071. 2023. DOI: 10.48550/arXiv.2307.02071.

[11] PStat; Mike Strube PhD; Allan Kozlowksi PhD PT Keith Lohse PhD. *Applied Mixed-Effects Regresion Resources*. 2022. URL: https://keithlohse.github.io/mixed_effects_models/.

[12] Holger Schielzeth and Shinichi Nakagawa. *Nested by design: model fitting and interpretation in a mixed model era*. Vol. 4. 1. 2013, pp. 14–24. URL: https://doi.org/10.1111/j.2041-210x.2012.00251.x.

[13] J. Bruin. *newtest: command to compute new test @ONLINE*. Feb. 2011. URL: https://stats.oarc.ucla.edu/stata/ado/analysis/.

[14] John Fox and Sanford Weisberg. *Mixed-effects Models in R: An Appendix to An R Companion to Applied Regression, Second Edition*. Tech. rep. 2015. URL: https://socialsciences.mcmaster.ca/jfox/Books/Companion-2E/appendix/Appendix-Mixed-Models.pdf.

[15] *Mixed Model*. URL: https://en.wikipedia.org/wiki/Mixed_model.

[16] A. Agresti et al. *2. Random-Effects Modeling of Categorical Response Data*. Vol. 30. 2000, pp. 27 –80. DOI: 10.1111/0081-1750.t01-1-00075.

[17] Giora Simchoni and Saharon Rosset. *Integrating Random Effects in Deep Neural Networks*. Vol. abs/2206.03314. 2022. DOI: 10.48550/arXiv.2206.03314.

[18] Giora Simchoni and Saharon Rosset. *Using Random Effects to Account for High-Cardinality Categorical Features and Repeated Measures in Deep Neural Networks*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 25111–25122. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/d35b05a832e2bb91f110d54e34e2da79-Paper.pdf.

[19] D. Nott M.-N. Tran N. Nguyen and R. Kohn. *Bayesian Deep Net GLM and GLMM*. Vol. 29. 1. Taylor Francis, 2020, pp. 97–113. DOI: 10.1080/10618600.2019.1637747. eprint: https://doi.org/10.1080/10618600.2019.1637747. URL: https://doi.org/10.1080/10618600.2019.1637747.

[20] K. Nguyen and A. Montillo. *Adversarially-regularized mixed effects deep learning (ARMED) models for improved interpretability, performance, and generalization on clustered data*. Vol. abs/2202.11783. 2022.

[21] Ahlem Hajjem, F. Bellavance, and Denis Larocque. *Mixed-effects random forest for clustered data*. Vol. 84. 2010, pp. 1313 –1328. DOI: 10.1080/00949655.2012.741599.

[22] Skipper Seabold and Josef Perktold. *Statsmodels: Econometric and statistical modeling with Python*. 2010.

[23] Martin Gardner. *The Bells: versatile numbers that can count partitions of a set, primes and even rhymes*. Vol. 238. 5. May 1978, pp. 24–30. DOI: 10.1038/scientificamerican0578-24.

[24] Jenks George F. (1967). *The Data Model Concept in Statistical Mapping*. 7, pp. 186–190.

[25] J. B. MacQueen. *Some Methods for classification and Analysis of Multivariate Observations*. Vol. 1. 1967, pp. 281–297.

[26] Lloyd S. Shapley. *A Value for n-Person Games*. Ed. by H. Kuhn and A. Tucker. Princeton: Princeton University Press, 1953, pp. 307–317. DOI: 10.1515/9781400881970-018.

[27] Gregory Belenky et al. *Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: a sleep dose-response study*. Vol. 12. 2003, pp. 1–12.

[28] Vincent Arel-Bundock. *Sleep Study Data from lme4 Package*. https://github.com/vincentarelbundock/Rdatasets/blob/master/csv/lme4/sleepstudy.csv.

[29] C. Lauridsen, S. Højsgaard, and M.T.C. Sørensen. *Influence of Dietary Rapeseed Oil, Vitamin E, and Copper on Performance and Antioxidant and Oxidative Status of Pigs*. Vol. 77. 1999, pp. 906–916.

[30] Vincent Arel-Bundock. *Dietox Dataset from geepack in R*. https://github.com/vincentarelbundock/Rdatasets/blob/master/csv/geepack/dietox.csv. 2023.