ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of Statistical Sciences "Paolo Fortunati"

Second Cycle Degree/Master in Statistical Sciences
Curriculum Official Statistics

Enhancing Automatic Editing with Machine Learning:

A Collaboration with Statistics Netherlands for Business Survey Statistics

Presented by:
Sona Yavrumyan
ID: 00001136875

Supervisor:
Prof. Stefano Lodi
Co-supervisor:
Dr. Sander Scholtus

Academic Year 2024/2025

Session I

Acknowledgements

I would like to express my sincere gratitude to all those who contributed to the realization of this thesis.

First and foremost, I am thankful to Statistics Netherlands (CBS) for providing me with the opportunity to carry out my internship in an environment rich in expertise and intellectual engagement. I am especially grateful to my supervisor at CBS, Dr. Sander Scholtus, whose mentorship, patience, and insightful guidance were essential to this work. I also wish to deeply thank Dr. Mark van der Loo, whose instrumental efforts in arranging this internship and whose unwavering dedication and encouragement have been truly inspiring throughout this process.

I am equally grateful to my supervisor at the University of Bologna, Professor Stefano Lodi, for academic oversight and guidance.

I am also grateful to the EMOS Master's Programme Board, with special thanks to Professor Elisabetta Carfagna, for the commitment to academic excellence and for the organization of a series of specialized initiatives and professional events within Eurostat that substantially advanced my knowledge and practical understanding of official statistics.

Finally, I would like to express my deep appreciation to my family and friends, whose unwavering support and understanding have been invaluable throughout this journey.

Abstract

A central challenge in the automatic editing of business survey data is the subjective assignment of reliability weights within the Generalized Fellegi-Holt (GFH) framework. This thesis addresses this problem by developing a data-driven methodology that uses machine learning to derive weights empirically, based on a collaboration with Statistics Netherlands (CBS).

The methodology uses historical manually corrected data to discover pairwise and higher-order co-editing patterns via association analysis, frequent itemset mining, and log-linear models. A hierarchical absorption analysis is introduced to define a non-redundant set of generalized edit operations. Subsequently, logistic regression models predict the probability of these error patterns occurring, conditional on business characteristics. These probabilities are transformed into dynamic, record-specific reliability weights.

Evaluation results demonstrate that the proposed dynamic and hierarchical weighting schemes consistently outperform static, predefined baselines. The most advanced model achieves the highest error detection rate (F1 Score), particularly for complex errors, and reveals a trade-off between error coverage, precision, and computational cost. This research provides a functional framework for integrating predictive modeling into statistical data editing, enhancing the accuracy and objectivity of such data processing pipelines.

Keywords: Automatic Data Editing, Machine Learning, Error Localization, Fellegi-Holt Paradigm, Reliability Weights.

Contents

Chapter 1

Introduction

National Statistical Institutes (NSIs), including Statistics Netherlands (CBS), are responsible for producing reliable and objective statistical information that serves governments, researchers, and the public. NSIs depend on data collected from various sources, such as business surveys and administrative registers. However, these data sources frequently contain errors, including measurement inaccuracies and logical inconsistencies, that can undermine the quality of statistical outputs. To address these issues, NSIs implement multi-stage data editing pipelines. While this process was traditionally conducted entirely through manual methods, the exponential growth in data volume has driven the adoption of automated systems. Nevertheless, manual editing is still applied to a subsample of cases. In practice, a strategy called "selective editing" (de Waal et al., 2011) is employed to ensure that manual effort is concentrated on identifying and correcting only the most influential errors. The manually-edited data generated from this selective editing process serves as the foundation for developing and evaluating the methodological approaches presented in this thesis.

A common approach in automatic editing is the paradigm introduced by Fellegi and Holt (1976). Their model formulates editing as a constrained optimization problem. Given a set of rules defining valid relationships between variables (e.g., turnover = revenue − costs), the objective is to modify the minimum number of values in a record to satisfy all rules. Each variable can be assigned a reliability weight reflecting its perceived accuracy. This leads to what is known as the error localization problem: identifying the smallest (or least weighted) subset of fields in a record that must be changed so that the resulting record satisfies all specified edit constraints. The solution to this problem is a corrected version of the record that restores consistency while minimizing the total cost of the corrections. The original Fellegi-Holt paradigm assumes that errors occur independently in individual fields and does not accommodate structured errors involving multiple variables, such as value swaps or systematic sign errors.

Generalizations of the paradigm address this limitation. The work of Scholtus (2016) and

Daalmans and Scholtus (2018) introduces *generalized edit operations*, which allow pre-specified operations to modify multiple fields in a structured manner. Each operation is assigned a reliability weight, and the optimization goal is to find the combination of operations that resolves all inconsistencies at the lowest total cost. This generalization allows the framework to model more realistic error patterns, but introduces the challenge of defining a suitable set of operations and assigning appropriate reliability weights. The selection and weighting of these operations has traditionally relied on domain expertise, an approach that is difficult to scale and may introduce inconsistencies.

This thesis presents a data-driven methodology built on an existing error localization algorithm developed at CBS. As part of the research, historical data was analyzed to compare raw survey responses with their manually corrected versions, identifying frequently co-occurring errors. Statistical association measures and network analysis were applied to extract these patterns, which were formalized into a set of generalized edit operations. To assign reliability weights to these operations, predictive models were developed to estimate the probability of each error pattern occurring in a given record. Several models were developed using machine learning, including adaptive systems that dynamically assign weights based on predicted error probabilities. These were compared to existing static-weight methods, which served as benchmarks. For the adaptive models, logistic regression was used to estimate probabilities from observable business characteristics (e.g., business size, legal form). The dynamic weighting system was further extended to incorporate a hierarchical scheme designed to model dependencies between related errors. Within this scheme, the weighting mechanism was designed to dynamically adjust an operation's reliability based on the presence of other correlated errors in the same record. The resulting probabilities from the chosen model were transformed into reliability weights for the error localization algorithm. The systems were applied to historical business survey data to evaluate the performance of each model, quantifying the trade-offs between model complexity, correction accuracy, and processing efficiency.

The remainder of the thesis is structured as follows. Chapter 2 reviews the theoretical foundations of automatic editing and the Fellegi-Holt paradigm. Chapter 3 details the methodology for pattern extraction and probability modeling. Chapter 4 presents the implementation, experimental design, and Chapter 5 - the evaluation results. Finally, Chapter 6 and 7 present limitations and suggestions for future research, and conclusions respectively.

Chapter 2

Background and Literature

This chapter presents the formal framework for automatic error localization in statistical data. The classical Fellegi-Holt paradigm is examined first, including its algorithm for generating implied edits and the structure of edit rules such as linear, conditional, and ratio edits. The text then introduces the Generalized Fellegi-Holt (GFH) paradigm, which is central to the thesis. This section provides its formal definition, a typology of generalized edit operations based on error mechanisms, and an analysis of its algorithmic complexities, including order-dependency. A Mixed-Integer Programming (MIP) formulation is offered as a solution, with mathematical techniques for linearizing non-linear operations (e.g., value swaps). The chapter concludes with an analysis of reliability weights and their function in the error localization process, providing the basis for the machine learning-based methods for weight estimation developed in the thesis's subsequent chapters.

2.1 The Classical Fellegi-Holt Paradigm

The work of Fellegi and Holt (1976) introduced one of the first comprehensive and mathematically rigorous frameworks for automatic data editing and imputation to be widely applied in official statistics. This contribution established an optimization-based methodology that has accompanied data quality assurance in statistical agencies. The paradigm addresses the problem of logical inconsistencies within a dataset by identifying a minimal set of alterations to the observed values required to restore conformity with a predefined system of constraints.

2.1.1 Formal Problem Definition

Let a data record consist of p variables, represented by a vector $x = (x_1, \ldots, x_p)$. The integrity of this record is assessed against a set of **edit rules**, denoted by $\mathcal{E}$. These rules collectively define the feasible region for a logically correct or consistent record. A record x is deemed consistent if and only if it satisfies every rule in $\mathcal{E}$. When a record is found to be inconsistent (i.e., it violates one or more edit rules), the Fellegi-Holt paradigm focuses on the *error localization problem*: identifying the specific subset of fields within the record that are most likely to be erroneous and whose modification would restore consistency.

To guide this identification process, each field x_i is typically assigned a positive reliability weight, $w_i > 0$. This weight quantifies the confidence in the correctness of the observed value x_i; a higher weight implies higher confidence and thus a greater "cost" associated with designating x_i as erroneous. A binary indicator vector $z \in \{0, 1\}^p$ is introduced, where $z_i = 1$ if field x_i is to be changed (imputed), and $z_i = 0$ if x_i is to be considered correct and left unchanged. The optimization problem at the heart of the Fellegi-Holt paradigm is then to find the indicator vector z that minimizes the sum of reliability weights of the fields designated for imputation, subject to the crucial condition that the record can indeed be made consistent by modifying only these designated fields:

$$\min_{z \in \{0,1\}^p} \left\{ \sum_{i=1}^{p} w_i z_i \quad \middle| \quad \exists y \text{ such that } y \text{ satisfies } \mathcal{E} \text{ and } y_i = x_i \text{ for all } i \text{ where } z_i = 0 \right\} \tag{2.1}$$

The underlying idea, often referred to as the "principle of minimum change," is that errors are relatively rare events. Therefore, the most plausible correction is one that requires modifying the fewest (or least reliable, in the weighted case) number of data items. If all $w_i = 1$, the problem reduces to finding the smallest number of fields to change. The solution to this error localization problem, z^*, identifies the fields that need to be imputed; the actual imputation of new, consistent values is a subsequent step in the editing process (Pannekoek et al., 2013).

2.1.2 The Structure and Specification of Edit Rules

The efficacy of the Fellegi-Holt paradigm heavily depends on the precise and comprehensive specification of the edit rules in $\mathcal{E}$. These rules encapsulate domain knowledge about the relationships between variables. In the context of business statistics, common types of edit rules include:

- **Linear Edits (Balance and Inequality Edits):** These are the most frequently encountered type of edit for numerical data. They express arithmetic relationships as linear combinations of variables, which can be written compactly using matrix notation. In this form, a

matrix A contains constant coefficients for the variables in x, while b and c are vectors containing the constant terms of the constraints. These edits are typically categorized as:

  - Balance edits are typically of the form $\sum_j a_{ij}x_j = c_i$ or, in matrix notation, $A_{\text{eq}}x = c$. A common example is an accounting identity, such as Total Assets = Total Liabilities + Equity, which translates to $x_{\text{Assets}} - x_{\text{Liabilities}} - x_{\text{Equity}} = 0$.
  - Inequality edits take the form $\sum_j a_{ij}x_j \leq b_i$ (or $\geq$), or $A_{\text{ineq}}x \leq b$. Examples include non-negativity constraints ($x_j \geq 0$), or logical bounds, e.g., *Number of Part-Time Employees ≤ Total Number of Employees*

- **Ratio Edits:** These are particularly prevalent in economic and business statistics, used to check the plausibility of relationships between financial or operational variables. A ratio edit constrains the ratio of two variables, x_i and x_j, to lie within a specified range, e.g., $L \leq x_i/x_j \leq U$. Such edits are inherently non-linear. For integration into linear programming-based solution frameworks, they often require linearization. Assuming $x_j > 0$ (a common pre-condition often checked by another edit), the ratio edit can be transformed into a pair of linear inequality edits: $L \cdot x_j - x_i \leq 0$ and $x_i - U \cdot x_j \leq 0$. If x_j can be zero or negative, more complex case-based linearization or alternative solution methods may be needed (de Waal et al., 2011).

- **Conditional Edits (Logical Edits):** These rules stipulate that a certain constraint (the consequent) must hold if a specific condition (the antecedent) is met. They often take the logical form of an implication, $P(x) \implies Q(x)$, which is read as "if $P(x)$ is true, then $Q(x)$ must also be true". For example, a rule might state that if a business reports zero turnover, then its total costs must also be zero (i.e., IF $x_{\text{turnover}} = 0$ THEN $x_{\text{costs}} = 0$). Another example could be that if a business is active, its employee count must be positive (i.e., IF $x_{\text{status}} =$ `'Active'` THEN $x_{\text{employees}} > 0$). Handling conditional edits in an optimization framework typically involves converting them into equivalent unconditional linear constraints, often by introducing binary indicator variables that represent the logical state of the antecedent. For instance, the rule IF $x_1 > 0$ THEN $x_2 > 0$ might be linearized using techniques discussed by, e.g., Williams (2013) for logical conditions in mathematical programming.

- **Categorical Edits:** While this thesis focuses on numerical business data, it is worth noting that edit rules can also apply to categorical variables (e.g., $x_{\text{country\_code}} =$ `'NL'` $\implies$ $x_{\text{currency\_code}} =$ `'EUR'`). These often involve set-based restrictions or logical dependencies between categories.

The accurate formulation and translation of these rule types into a consistent mathematical representation (typically a system of linear equations and inequalities) is a preparatory step for automatic editing.

2.1.3 Algorithmic Solution via Generation of Implied Edits

The original algorithmic approach proposed by Fellegi and Holt (1976) for solving the error localization problem involves leveraging the full logical structure of the edit system through the concept of *implied edits*. An implied edit is a constraint that is not explicitly stated in the initial set $\mathcal{E}$ but is a necessary logical consequence of the rules within $\mathcal{E}$. For example, if the explicit rules include $x_1 + x_2 = x_3$ and $x_3 \geq 10$, then an implied edit is $x_1 + x_2 \geq 10$.

The classical Fellegi-Holt algorithm first performs a one-time expansion of the initial rules into a complete set of edits. This derivation is achieved through an exhaustive procedure of logical deduction; for example, Fourier-Motzkin (de Waal et al., 2011) elimination can be used to generate all implied constraints from a system of linear inequalities (Williams, 2013). Once this complete set is established, the error localization for any given inconsistent record identifies all edits violated by that record. The principle that at least one variable in every violated edit must be changed transforms the problem into an instance of the minimum weight set-covering problem. In this formulation, each variable is an element with a cost equal to its reliability weight, and each violated edit defines a subset of these variables; the goal is to select a minimum-weight collection of variables that "covers" every such subset.

Although this approach guarantees an optimal solution to (2.1), its practical application is limited by the computational cost of generating the complete set of implied edits. The number of implied edits can grow exponentially with the number of variables and original rules, rendering this preliminary step computationally infeasible for many problems (Garfinkel et al., 1986). This computational bottleneck motivated research into alternative solution methods, including direct Mixed-Integer Programming (MIP) formulations for the classical problem (de Jonge and van der Loo, 2014) and for its generalizations.

2.2 The Generalized Fellegi-Holt Paradigm

In many practical data editing scenarios, particularly with complex survey instruments or administrative data sources, errors are often not isolated, random occurrences. Instead, they can be systematic (e.g., consistent misinterpretation of a question) or structural (e.g., swapping values between two related fields). The classical Fellegi-Holt paradigm implicitly assumes that errors in different fields are independent events. This assumption is reflected in its focus on changing individual fields, which renders it inherently limited in its ability to model and correct complex error mechanisms that violate this principle of independence.

Recognizing this limitation was a primary motivation for the development of the Generalized Fellegi-Holt (GFH) paradigm, most notably elaborated by Scholtus (2016). The GFH framework

overcomes this by explicitly relaxing the independence assumption, allowing for predefined "operations" that can model and correct multiple fields simultaneously.

The paradigm introduces the concept of the generalized **edit operation**. An edit operation is a predefined, structured transformation that can modify one or more fields of a data record simultaneously. This moves beyond the simple "change value of field x_i" action of the classical model. Formally, an edit operation o_k can be defined as a function that maps an original record x and, potentially, a vector of free parameters θ_k (which might be determined during a subsequent imputation phase) to a new, modified record x'. The operation is often more conveniently represented by the change vector, $\Delta_k(x; \theta_k)$, that it induces. The corrected record resulting from the application of this single operation is then:

$$x' = x + \Delta_k(x; \theta_k) \tag{2.2}$$

Notably, Δ_k can specify simultaneous changes to multiple components of x, and these changes can be interdependent or based on the original values in x. The task of error localization within the GFH framework is to identify which of these predefined operations should be applied to an inconsistent record to restore consistency. The determination of optimal values for any free parameters θ_k (e.g., the amount to be transferred in a "transfer" operation) is typically deferred to the imputation step, which follows error localization.

2.2.1 A Typology of Generalized Edit Operations

The GFH paradigm models a wide variety of realistic error types that are commonly encountered by data editors. Below are several illustrative examples of such operations, many of which mirror the complex corrections human editors perform:

- **Value Swap (Interchange of Values):** This operation corrects for the erroneous interchange of values between two specific fields, say x_i and x_j. The change vector effectively sets $x'_i = x_j$ and $x'_j = x_i$, while leaving other fields unchanged. This is a common error in data entry from forms where columns might be misread.

- **Sign Flip (Sign Correction):** This operation addresses a systematic sign error in a single field, $x_i \rightarrow -x_i$. This is particularly relevant for financial data where debits and credits, or profits and losses, might be reported with an incorrect sign.

- **Transfer of Amount:** This models the incorrect allocation of a certain amount α between two (or more) fields. For instance, if an amount α was erroneously reported in field x_i instead of x_j, the operation would be $x'_i = x_i - \alpha$ and $x'_j = x_j + \alpha$. The parameter α itself

might be a free parameter to be determined, or it could be fixed if the nature of the error implies a specific amount.

- **Systematic Proportional Error (e.g., Rounding or Unit Error):** This operation corrects for values that were systematically misreported by a common factor, such as reporting in thousands of euros instead of euros ($x_i \rightarrow x_i/1000$), or vice-versa. This can affect multiple related fields simultaneously. The change would be $\Delta_k(x)_i = (\lambda - 1)x_i$ for each affected field i, where λ is the scaling factor (e.g., $1/1000$).
- **Systematic Addition/Subtraction Error:** A constant amount might be incorrectly added to or subtracted from a group of related variables.
- **Copying Error (Duplication):** A value from one field x_i might have been incorrectly copied to another field x_j, where x_j should have had a different value. The operation would restore x_j to some valid imputed value while potentially marking x_i (or the copying action itself) as the source of the error.
- **Deletion of a Dependent Figure:** If a primary figure is reported as zero or missing, a dependent figure that should also be zero/missing might still be reported. An operation could simultaneously correct both.

The set of defined operations, O, for a particular application is usually domain-specific and is compiled based on expert knowledge of common error types and analysis of historical manual edits. The classical Fellegi-Holt action of imputing a single field x_i with an arbitrary new value can be seen as a special case of a generalized operation where $\Delta_k(x)_i = \theta_{ki} - x_i$ (where θ_{ki} is the new value) and $\Delta_k(x)_j = 0$ for $j \neq i$.

2.2.2 The Generalized Error Localization Problem

With the introduction of this richer set of generalized operations, the error localization problem is fundamentally redefined. Each potential edit operation $o_k \in O = \{o_1, \ldots, o_K\}$ is assigned a reliability weight w_k^{EO} (where EO stands for Edit Operation). This weight reflects the presumed likelihood or cost of this specific error mechanism occurring. The objective is to find a sequence of operations $(o_{k_1}, o_{k_2}, \ldots, o_{k_m})$, drawn from the available set O, that transforms the original inconsistent record x into a consistent record $x^{(m)}$, while minimizing the total sum of weights of the applied operations:

$$\min_{(o_{k_1},\ldots,o_{k_m})} \sum_{j=1}^{m} w_{k_j}^{EO} \quad \text{such that} \quad x^{(m)} = (o_{k_m} \circ \cdots \circ o_{k_1})(x) \text{ satisfies } \mathcal{E} \tag{2.3}$$

where $\circ$ denotes function composition (i.e., $o_2 \circ o_1(x) = o_2(o_1(x))$).

The generalized error localization problem as defined in (2.3) is computationally more complex than its classical counterpart. The primary challenge is the issue of **order-dependency**. The effect of applying a sequence of generalized edit operations can, and often does, depend critically on the order in which those operations are executed. For example, applying a "sign flip" to x_i and then a "transfer of amount α from x_i to x_j" will generally yield a different result than performing the transfer first and then flipping the sign of the (already modified) x_i.

Order-dependency means that a simple search for an optimal *set* of operations is insufficient; one must, in principle, search for an optimal *sequence*. This leads to a combinatorial explosion in the search space. If there are K potential edit operations and one considers corrective sequences of length up to m, the number of possible ordered sequences to evaluate can be on the order of K^m. Exploring this vast space, which grows exponentially with the length of the sequence, to find the true minimum-weight sequence that achieves consistency is generally computationally intractable for realistic problem sizes. Scholtus (2016) proposed an algorithm for the generalized problem, noting its computational limitations, particularly due to the need to handle implied edits in a generalized context and the order-dependency issue. These computational challenges motivated the search for a more efficient and scalable approach, which led to the Mixed Integer Programming (MIP) formulation.

2.3 A Computationally Feasible Solution through the MIP Formulation

A key development that enabled the application of the Generalized Fellegi-Holt (GFH) paradigm to practical, large-scale problems was its formulation as a Mixed-Integer Program (MIP), notably detailed by Daalmans and Scholtus (2018). Mixed-Integer Programming is a powerful mathematical optimization framework where some decision variables in the problem are constrained to take on only integer values (e.g., binary 0 or 1 values to represent discrete choices), while other decision variables can assume continuous values. This versatile structure allows the MIP approach to effectively manage, and largely circumvent, the debilitating order-dependency issue inherent in the sequential application of generalized edit operations. This is achieved through the incorporation of specific simplifying, yet operationally pragmatic, assumptions regarding the application order of corrections. Furthermore, the MIP formulation leverages the advanced capabilities of general-purpose optimization solvers for its efficient computation, making the GFH paradigm tractable for NSI production environments.

2.3.1 The Complete Mixed-Integer Programming Model

The MIP model for the GFH problem introduces binary decision variables to represent the discrete choice of whether to apply each potential corrective action. Let the variables and parameters be defined as follows (Daalmans and Scholtus, 2018):

- $\delta_i^{\mathrm{FH}} \in \{0, 1\}$: This binary variable is equal to 1 if a standard Fellegi-Holt style imputation (i.e., replacing the original value of variable i with an arbitrary new value) is applied to variable i (for $i = 1, \ldots, p$), and 0 otherwise.
- $\delta_k^{\mathrm{EO}} \in \{0, 1\}$: This binary variable is equal to 1 if a specific generalized edit operation o_k (from a predefined set O of K such operations, so $k = 1, \ldots, K$) is applied, and 0 otherwise.
- x_i': This continuous variable represents the value of the i-th field (for $i = 1, \ldots, p$) in the data record after all selected corrections have been applied. The vector of these corrected values is denoted by $\mathbf{x}'$.
- α_{kr}: This continuous variable represents the r-th free parameter (where $r = 1, \ldots, m_k$, with m_k being the number of free parameters for operation o_k) associated with the generalized edit operation o_k. These parameters allow for flexibility in certain operations, such as the amount to be transferred in a transfer operation. The vector of these parameters for a given operation k is $\boldsymbol{\alpha}_k$.

Let $\mathbf{x}^0 = (x_1^0, \ldots, x_p^0)$ be the vector of original observed (preliminary) values for the p variables in the record. The objective function of the MIP is to minimize the total weighted cost of all applied corrections. This cost encompasses both the standard FH-style imputations and the generalized edit operations:

$$\min_{\mathbf{x}', \boldsymbol{\delta}^{\mathrm{FH}}, \boldsymbol{\delta}^{\mathrm{EO}}, \boldsymbol{\alpha}} \sum_{i=1}^{p} w_i^{\mathrm{FH}} \delta_i^{\mathrm{FH}} + \sum_{k=1}^{K} w_k^{\mathrm{EO}} \delta_k^{\mathrm{EO}} \tag{2.4}$$

This minimization is subject to a system of linear constraints that collectively model the logic of the editing process, ensuring that the final record is consistent and that operations are applied

correctly:

$$\mathbf{A}x' + \mathbf{b} \odot \mathbf{0} \qquad \text{(Edit rule satisfaction by } x') \tag{2.5}$$

$$x_i^0 - C_i^{\text{FH}} + C_i^{\text{EO}} \leq x_i' \leq x_i^0 + C_i^{\text{FH}} + C_i^{\text{EO}} \qquad (\forall i = 1, \ldots, p) \tag{2.6}$$

$$C_i^{\text{FH}} = M\delta_i^{\text{FH}} \qquad (\forall i = 1, \ldots, p) \tag{2.7}$$

$$C_i^{\text{EO}} = \sum_{k=1}^{K} I_{ki}^{\text{EO}}(C_{ki}^D + C_{ki}^V) \qquad (\forall i = 1, \ldots, p) \tag{2.8}$$

$$C_{ki}^D = \delta_k^{\text{EO}} \left(\sum_{j=1}^{p} t_{kij} x_j^0 + c_{ki} - x_i^0 \right) \qquad (\forall k = 1, \ldots, K; \forall i = 1, \ldots, p) \tag{2.9}$$

$$C_{ki}^V = \delta_k^{\text{EO}} \sum_{r=1}^{m_k} s_{kir}\alpha_{kr} \qquad (\forall k = 1, \ldots, K; \forall i = 1, \ldots, p) \tag{2.10}$$

$$-M\delta_k^{\text{EO}} \leq \alpha_{kr} \leq M\delta_k^{\text{EO}} \qquad (\forall k = 1, \ldots, K; \forall r = 1, \ldots, m_k) \tag{2.11}$$

$$\sum_{k=1}^{K} \delta_k^{\text{EO}} I_{ki}^{\text{EO}} \leq 1 \qquad (\forall i = 1, \ldots, p) \tag{2.12}$$

Here, x_i^0 is the original observed value of variable i. I_{ki}^{EO} is an indicator parameter (0 or 1) which is 1 if variable i is (potentially) affected by generalized edit operation o_k, and 0 otherwise. The parameters t_{kij} and c_{ki} are coefficients that define the fixed part of operation o_k's effect on variable i (this effect is calculated relative to the original values x_j^0 and x_i^0). The parameters s_{kir} are coefficients for the variable part of the operation's effect, which depends on the free parameters α_{kr}. Finally, M represents a sufficiently large positive constant, commonly used in MIP formulations (the "Big M" method).

The constraints define the rules for applying edit operations and enforce the consistency of the resulting record.

- **Constraint (2.5):** Ensures that the final, corrected record $\mathbf{x}'$ must satisfy all defined edit rules (e.g., $Ax' \leq b$). The $\odot$ represents the appropriate relational operators for each rule.
- **Constraint (2.6):** Defines the relationship between the original value x_i^0, the corrected value x_i', and the changes due to FH imputation (C_i^{FH}) and special edit operations (C_i^{EO}). If $\delta_i^{\text{FH}} = 1$, then $C_i^{\text{FH}} = M$, allowing x_i' to take any value within the broad range $x_i^0 + C_i^{\text{EO}} \pm M$. If $\delta_i^{\text{FH}} = 0$, then $C_i^{\text{FH}} = 0$, which forces $x_i' = x_i^0 + C_i^{\text{EO}}$. This signifies that x_i' is the original value x_i^0 plus the net change from any active EOs, or it is freely imputed (subject to edit rules) if an FH-imputation is chosen for variable i.
- **Constraint (2.7):** This is a standard "big M" constraint linking the continuous "correction allowance" C_i^{FH} to the binary decision δ_i^{FH}. If $\delta_i^{\text{FH}} = 0$, $C_i^{\text{FH}} = 0$. If $\delta_i^{\text{FH}} = 1$, $C_i^{\text{FH}} = M$.
- **Constraint (2.8):** Defines C_i^{EO} as the sum of net changes to the original value x_i^0 from all

active generalized edit operations (EOs) that affect variable i.

- **Constraint (2.9):** Defines C_{ki}^{D} as the fixed part of the change to variable i resulting from operation o_k (calculated relative to its original value x_i^0), activated only if $\delta_k^{\text{EO}} = 1$. This component explicitly depends on the original values x_j^0 of variables in the record.
- **Constraint (2.10):** Defines C_{ki}^{V} as the variable part of the change to variable i resulting from operation o_k, dependent on the free parameters α_{kr} and activated only if $\delta_k^{\text{EO}} = 1$.[1]
- **Constraint (2.11):** A "Big M" constraint ensuring that the free parameters α_{kr} for an operation o_k can only be non-zero (and take values within $\pm M$) if that operation o_k is selected ($\delta_k^{\text{EO}} = 1$). If $\delta_k^{\text{EO}} = 0$, then α_{kr} is forced to be 0.
- **Constraint (2.12):** This constraint ensures that each variable x_i can be affected by at most one specialized edit operation (in addition to potentially an FH-imputation).

Key assumptions made by Daalmans and Scholtus (2018) to manage order-dependency and make the MIP tractable are:

1. *All special edit operations are considered to be applied to the original observed record p.* This means the fixed part of an operation C_{ki}^{D} depends on p, not on values already modified by other EOs, following from Constraint 2.12.
2. *In an optimal solution, Fellegi-Holt (FH) imputation operations are always applied after any special edit operations.*

Under these assumptions, the MIP treats operations as if EOs transform the original record $\mathbf{x}^0$ to an intermediate state, which FH imputations then finalize to $\mathbf{x}'$. This allows for finding optimal EOs and FH imputations without exploring all possible operational sequences.

2.3.2 Linearization of Inherently Non-Linear Operations in the MIP Framework

A practical challenge in formulating the GFH problem as an MIP is that some desirable generalized edit operations, such as a "value swap" between variables x_i and x_j, are inherently

[1] This formulation explicitly defines C_{ki}^{D} as the deviation from the original value x_i^0 caused by the data-dependent components of operation o_k, and C_{ki}^{V} as the deviation caused by its free parameters α_{kr}. The sum of these deviations, aggregated in C_i^{EO}, represents the total net change to x_i^0 from all active generalized edit operations. This total change is then applied to x_i^0 in constraint (2.6) to determine x_i' (if $\delta_i^{\text{FH}} = 0$). While Daalmans and Scholtus (2018) their intermediate terms leading to the final value slightly differently (where their C_{ki}^{D} component, for instance, already incorporates $-x_i^0$), the resulting corrected value x_i' is mathematically equivalent under both modeling choices when the operations are active.

non-linear. If operation o_k represents a swap, the change to x_i would be $\Delta_{ki} = x_j^0 - x_i^0$, and to x_j would be $\Delta_{kj} = x_i^0 - x_j^0$. If these changes were to be directly incorporated into the model such that a term like $\delta_k^{\text{EO}} \cdot x_j^0$ arises, and if x_j^0 were itself a decision variable rather than a fixed parameter, this product of a binary variable and a continuous variable would be non-linear.

Standard techniques exist for linearizing such products (Williams, 2013). For a swap operation o_k affecting x_i and x_j (using original values x_i^0, x_j^0): The intended effect is: if $\delta_k^{\text{EO}} = 1$, then $x_i' = x_j^0$ and $x_j' = x_i^0$. If $\delta_k^{\text{EO}} = 0$, then $x_i' = x_i^0$ and $x_j' = x_j^0$ (assuming no other operations modify these variables and $\delta_i^{\text{FH}} = \delta_j^{\text{FH}} = 0$).

This transformation (when $\delta_k^{\text{EO}} = 1$) implies specific net changes that contribute to C_i^{EO} for variable i (namely, $x_j^0 - x_i^0$) and to C_j^{EO} for variable j (namely, $x_i^0 - x_j^0$). These specific change values are then incorporated into the fixed part of the operation, C_{ki}^D and C_{kj}^D respectively, within the structure of constraint (2.9). For instance, for variable x_i, the term $\left(\sum_{l=1}^{p} t_{kil} x_l^0 + c_{ki}\right)$ within C_{ki}^D would be structured to equal x_j^0. Since x_j^0 is a known original value (a parameter), the product $\delta_k^{\text{EO}} \cdot$ (terms involving x_j^0 and x_i^0) remains linear.

However, if a more general non-linearity arises, such as needing to model $v = \delta \cdot y$, where $\delta \in \{0, 1\}$ is a binary decision variable and y is a continuous decision variable with known bounds $L \leq y \leq U$, then specific linearization constraints are required for the new variable v:

$$L\delta \leq v \leq U\delta$$
$$y - U(1 - \delta) \leq v \leq y - L(1 - \delta)$$

This set of linear constraints enforces $v = y$ if $\delta = 1$, and $v = 0$ if $\delta = 0$. Similar constructs are used for other non-linearities involving products of binary and continuous decision variables, ensuring the entire problem remains within the scope of standard MIP solvers. The general structure of C_{ki}^D and C_{ki}^V in the model (constraints (2.9)-(2.10)) is designed to directly incorporate coefficients that define such linear (or linearized) operations.

2.4 The Theory and Practice of Reliability Weights

The successful application and performance of the Fellegi-Holt framework, in both its classical and generalized forms, are critically dependent on the appropriate specification of the reliability weights (w_i^{FH} for single-field imputations and w_k^{EO} for generalized edit operations). These weights guide the optimization process in selecting the "most plausible" set of corrections among potentially many alternatives that could restore consistency.

2.4.1 The Statistical Interpretation of Weights and Its Implications

The interpretation of the Fellegi-Holt paradigm as an approximate maximum-likelihood procedure was first proposed by Liepins (1980) for the classical case, which was later extended to the Generalized Fellegi-Holt paradigm by Scholtus (2016). It was demonstrated that if errors in individual variables x_i are assumed to occur independently with a known probability p_i, then the objective of minimizing the sum of weighted changes $\sum w_i z_i$ is approximately equivalent to maximizing the posterior probability (or likelihood, under certain assumptions) of the chosen set of erroneous fields, provided the weights are set according to the log-odds ratio:

$$w_i = \log\left(\frac{1 - p_i}{p_i}\right) = logit(1 - p_i) \tag{2.13}$$

This formula implies that fields with a high probability of being erroneous ($p_i \to 1$) receive a low (or even negative, if $p_i > 0.5$) weight, making them "cheaper" to change. Conversely, fields with a low error probability ($p_i \to 0$) receive a high positive weight, making them "expensive" to change. This principle extends conceptually to the GFH paradigm: the weight w_k^{EO} for a generalized edit operation o_k should ideally be derived from an estimate of the probability, $\hat{p}_k$, that the specific error mechanism corresponding to o_k has occurred in the given record. Thus, $w_k^{EO} = \log((1 - \hat{p}_k)/\hat{p}_k)$. This statistical interpretation transforms weight setting from an ad-hoc or purely subjective exercise into a problem of statistical estimation and modeling. It provides a theoretical basis for developing data-driven methods to determine weights, which is a central focus of this thesis.

2.4.2 The Dominance Problem and Hierarchical Weighting Strategies

A practical challenge in assigning weights, especially in the GFH paradigm with a potentially large and overlapping set of defined operations, is the "dominance" problem. This occurs when the weight settings inadvertently cause the optimization algorithm to systematically prefer certain types of operations over others in a way that contradicts expert judgment or empirical evidence about error likelihoods. As an example, a very general operation (like "impute field x_i with an arbitrary value") might dominate a more specific, and potentially more plausible, operation (like "apply a sign flip to x_i") if the weight for the specific operation is not set sufficiently lower than that of the general one.

To prevent such undesirable dominance and ensure that the system selects the most parsimonious and contextually appropriate corrections, weights must often be set hierarchically. This means that the weight for a more specific or structured operation o_k should be less than the sum of weights of the simpler or more general operations that would be required to achieve a similar corrective effect on the data. For instance, to ensure a "value swap" operation $o_{\text{swap}(i,j)}$ is chosen

over independently imputing x_i and x_j, the weights should satisfy:

$$w^{EO}_{\text{swap}(i,j)} < w^{FH}_i + w^{FH}_j \tag{2.14}$$

Similarly, if operation o_A can achieve the same result as a combination of operations o_B and o_C, then ideally $w^{EO}_A < w^{EO}_B + w^{EO}_C$ if o_A is considered a more direct explanation. Establishing a consistent and comprehensive hierarchy of weights for a large set of potentially overlapping and interacting operations is a non-trivial task. It requires careful analysis of the relationships between operations and a clear understanding of the relative likelihoods of different error mechanisms (Scholtus, 2016; Daalmans and Scholtus, 2018). Failure to address weight dominance can lead to suboptimal or implausible automatic corrections.

2.4.3 Methodologies for Weight Estimation

The practical estimation of the error probabilities p_i (for FH) or p_k (for GFH operations), which are required to calculate the theoretically optimal weights using (2.13), represents a central component in the implementation of the Fellegi-Holt framework. Two distinct conceptual approaches to weight estimation can be identified.

- **Static Weights (Global Weights):** In this common approach, a single, fixed reliability weight is assigned to each variable (for FH) or to each type of generalized edit operation (for GFH). These weights are typically derived from aggregate historical data, such as the overall frequency of errors observed in a particular field across many past survey instances, or the historical frequency of a specific type of complex error (e.g., value swaps between two particular variables). Expert judgment may also play a significant role in setting these global weights, especially when historical data is scarce or deemed unreliable for certain error types. While relatively simple to implement, static weights have a major limitation: they ignore the fact that the probability of a specific error occurring can vary significantly depending on the characteristics of the particular data record being edited, or the context provided by other variables in that record. A weight defined in the following manner may not be optimal across a heterogeneous population of records.

- **Dynamic Weights (Record-Specific or Conditional Weights):** Here, the error probability, and consequently the reliability weight, is not fixed but is estimated dynamically for each specific data record under consideration. The probability $\hat{p}_k(x)$ (for operation o_k applied to record x) becomes a function of the observable features of that record. For example, a sign error in a financial variable might be more probable for small, newly established businesses than for large, well-established corporations. Similarly, a value swap might be more likely if the two variables involved have similar magnitudes or are adjacent

on a survey form. This dynamic estimation of error probabilities can be framed as a predictive modeling problem. Techniques from statistical modeling or machine learning, such as logistic regression, can be employed to build models that predict $\hat{p}_k(x)$ based on a set of predictor variables $r_1, \ldots, r_q$ derived from the record x (e.g., industry code, company size, values of related variables, specific patterns of edit failures). A logistic regression model would take the form:

$$\text{logit}(\hat{p}_k(x)) = \log\left(\frac{\hat{p}_k(x)}{1-\hat{p}_k(x)}\right) = \beta_0 + \beta_1 r_1(x) + \cdots + \beta_q r_q(x) \tag{2.15}$$

The coefficients β_j would be estimated from historical, manually edited data where corrections are known. The predicted $\hat{p}_k(x)$ from such a model would then be used to calculate a record-specific weight $w_k^{EO}(x)$ via (2.13). This approach allows the automatic editing process to adapt its assumptions about error likelihoods to the specific context of each individual record, potentially leading to more accurate and plausible corrections. The development and evaluation of dynamic weighting schemes using machine learning is which will be explored further in the thesis.

Beyond these, other considerations in weight setting include the potential use of Bayesian methods to incorporate prior beliefs about error probabilities, methods for handling uncertainty in probability estimates, and sensitivity analyses to understand the impact of weight variations on editing outcomes.

2.5 Summary and Identification of the Research Gap

To address certain limitations of the classical theory, the Generalized Fellegi-Holt (GFH) paradigm was developed as a more robust framework. The subsequent development of the Mixed-Integer Programming formulation for GFH, particularly by Daalmans and Scholtus (2018), has provided a computationally viable and robust pathway for implementing this generalized approach in demanding NSI production environments. This has opened up new possibilities for automating the correction of complex, structural errors that were previously difficult to handle systematically.

However, the practical effectiveness and the quality of corrections produced by the GFH framework, even with an efficient MIP solver, remain contingent on two key sets of inputs:

1. **A relevant and comprehensive set of generalized edit operations (o_k):** These operations must accurately reflect the actual error mechanisms that occur in the specific data domain being processed. If the defined operations are misaligned with common errors, or if

significant error types are omitted, the system's ability to find plausible corrections will be compromised.

2. **An accurate and well-calibrated set of reliability weights (w_k^{EO} and w_i^{FH}):** These weights are essential for guiding the optimization process towards the most likely error explanations.

While the literature strongly advocates for deriving these inputs empirically from data (e.g., from historical manual edits), and acknowledges the theoretical link between weights and error probabilities, it provides comparatively limited guidance on systematic, scalable, and robust methodologies for achieving this in practice. Specifically, there is a need for:

- Methods to systematically discover and define new, potentially complex, generalized edit operations directly from data, rather than relying solely on pre-existing expert knowledge.
- Advanced techniques for estimating error probabilities, particularly for creating dynamic, record-specific reliability weights that adapt to the context of individual data records.

This thesis aims to address precisely this research gap. It focuses on developing and evaluating a data-driven methodology that integrates techniques from statistical modeling and machine learning into the GFH framework. The proposed approach has two primary components:

1. **Discovering common complex error patterns from historical manually edited data:** This involves applying pattern mining or other relevant machine learning algorithms to identify recurring multi-field correction patterns, which can then be formalized as candidate generalized edit operations for the GFH system.
2. **Building predictive models to estimate error probabilities dynamically:** This involves using historical data to train models (such as logistic regression) that predict the probability of specific error patterns occurring in a record. These predictions will be based on the characteristics of the business record itself as well as the contextual nature of the error pattern (e.g., whether an edit operation is indicated individually, or as part of a pair, triplet, or larger group of concurrent corrections). These context-sensitive predicted probabilities are then transformed into reliability weights for use in the MIP-based error localization.

By deriving edit operations from empirical patterns and dynamically estimating reliability weights through predictive models, this thesis proposes methodology to improve automatic data editing. The goal is to enhance accuracy, objectivity, and efficiency in the data processing pipelines. Subsequent chapters will elaborate on these methodologies, their practical implementation, and the findings from their evaluation on business statistics data (particularly on the CBS "Structural Business Statistics Survey").

Chapter 3

Methodology for Discovering Error Patterns and Deriving Empirical Weights

This chapter details the data-driven methodology developed to enhance automatic data editing for the CBS "Structural Business Statistics Survey". The objective is to identify complex, multivariable error patterns from historical, manually corrected data, moving beyond static rule sets. Insights from this empirical investigation are used to construct an adaptive, predictive framework for producing reliability weights for the Generalized Fellegi-Holt (GFH) error localization system described in Chapter 2.

The analysis utilizes a pre-chosen sample of the "Structural Business Statistics Survey" dataset, comprising approximately 16,000 enterprises across seven size categories (small, medium, and large), with around 100 variables per record covering financial and operational metrics [1]. The dataset includes both raw survey responses and manually corrected versions, allowing for direct analysis of expert editing patterns across different business characteristics including size class, legal form, sector classification, and operational complexity. The analysis begins with chi-square tests to identify significantly co-edited variable pairs, examining error heterogeneity across different business types. These pairwise associations inform logistic regression models that estimate probabilities of co-editing patterns based on observable business characteristics (size, legal structure, and business date of formation). Subsequently, association rule mining using the Eclat algorithm identifies frequent higher-order variable combinations (triplets and quadruplets) from the most frequently edited variables. Log-linear models test for true interaction effects in these higher-order combinations, while a hierarchical absorption analysis determines which lower-order patterns are subsumed by higher-order patterns, enabling construction of non-redundant edit operation hierarchies. The probabilistic outputs from individual variable models, pairwise models, and higher-order interaction models generate dynamic,

[1] An illustrative example of the dataset structure and variable types can be found in Table A.1 in the Appendix.

record-specific reliability weights for the Mixed-Integer Programming (MIP) error localization solver (Section 2.3). The flowchart below summarizes this methodology (Figure 3.1), with detailed implementation discussed in the following subsections.

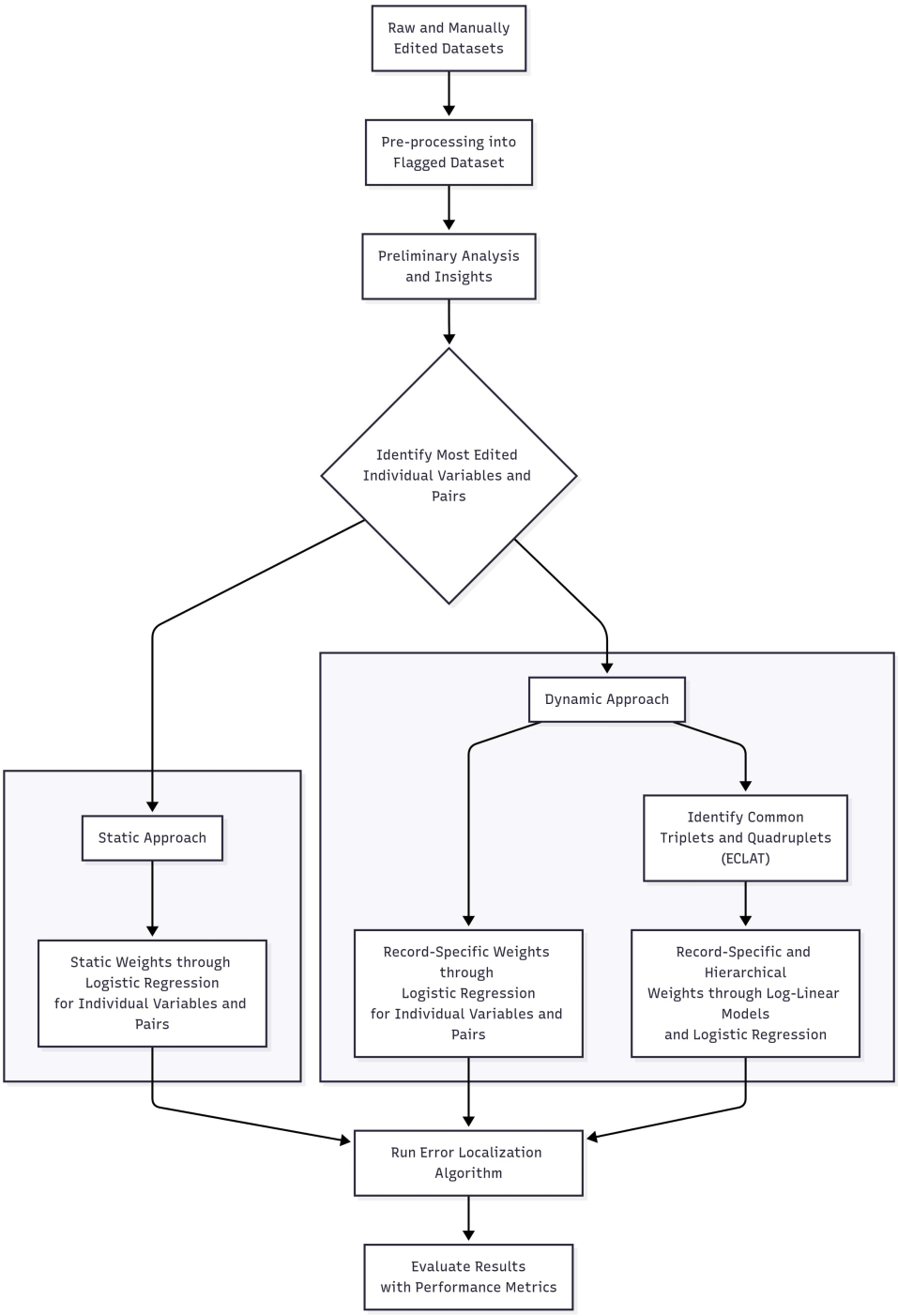


Figure 3.1: Workflow for Error Localization using Static and Dynamic Weights

3.1 Data Foundation

The research utilizes historical records from the CBS "Structural Business Statistics Survey". Data from two consecutive survey years, 2021 and 2022, were combined to ensure a representative sample.

For each survey year, two datasets are available for each responding business:

1. A *raw* dataset: verbatim responses as submitted.
2. An *edited* dataset: records post-CBS data processing, which includes automated checks and, where flagged, manual review and correction by domain experts. This methodology assumes that expert corrections serve as the definitive reference point for data quality and those manual corrections serve as the training target for predictive modeling.

Raw survey responses from manually edited records were aligned with their corrected counterparts using unique business and year identifiers, producing a matched dataset with pre- and post-correction values for each survey variable per record. This aligned dataset was transformed into a binary edit-flag matrix where each row represents a business-year observation and columns represent survey variables as binary indicators of manual correction (Table 3.1). For each survey variable v_i in record k, a flag v_{ik}^{flag} was computed:

$$v_{ik}^{\text{flag}} = \begin{cases} 1 & \text{if } v_{ik}^{\text{raw}} \neq v_{ik}^{\text{edited}} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where v_{ik}^{raw} is the original value and v_{ik}^{edited} is the manually corrected value.

A refinement was applied for missing data: a change between a missing value (NA) and a reported value of zero (or vice-versa) was not considered an edit. Variables entirely missing in both raw and edited versions were excluded. The resulting edit-flag matrix provides a machine-readable representation of the manual correction process. An illustrative example for a given variable A and B can be found in Table 3.1 [2].

Table 3.1: Illustrative structure of raw, edited, and flag values for selected variables

Business ID	**Year**	A_raw	A_edit	A_flag	B_raw	B_edit	B_flag
1001	2021	100	100	0	50	55	1
1002	2021	200	250	1	NA	0	0
1001	2022	110	110	0	60	60	0
1003	2022	NA	NA	0	30	0	1

[2]True variable names from the survey would replace placeholders.

3.2 Pairwise Association Analysis

The edit-flag matrix identifies survey variables frequently edited together, revealing statistically significant associations in editing behavior. Such pairs indicate violations of the classical Fellegi-Holt paradigm's assumption of independent errors.

Edit frequencies were computed for all survey variables as $f_i = \sum_k v_{ik}^{\text{flag}}$ and analysis was restricted to the 20 most frequently edited variables. This selection is due to both computational complexity and the practicality that domain experts concentrate their editing efforts on the most erroneous and influential variables. This ensures focus on variables with the highest contribution to manual editing and sufficient instances for statistical power in subsequent analyses.

To identify statistically significant co-editing relationships beyond simple co-occurrence, a pairwise association analysis was performed. For each unique pair of variables (v_a, v_b) from the top 20, a 2×2 contingency table was constructed, cross-tabulating their edit status (edited/not-edited).

A Pearson's Chi-square (χ^2) test of independence was applied to each table. The null hypothesis (H_0) states that the editing of v_a is independent of the editing of v_b. A statistically significant result (low p-value) suggests that the variables are co-edited more or less often than expected by chance. The χ^2 statistic is:

$$\chi^2 = \sum_{r,c} \frac{(O_{rc} - E_{rc})^2}{E_{rc}} \tag{3.2}$$

where O_{rc} is the observed frequency and E_{rc} is the expected frequency under H_0.

For the ${}_{20}C_2 = 190$ tests conducted, the Benjamini–Hochberg procedure (Benjamini and Hochberg, 1995) was applied to the raw p-values to control the False Discovery Rate (FDR). A co-editing relationship was deemed statistically significant if its adjusted p-value was below 0.05. This process yielded a list of 190 statistically significant variable pairs with associated edits. From these significant pairs, a final subset was selected based on practical relevance using logistic regression models fitted to predict co-editing probabilities based on business characteristics. A total of 50 pairs were retained based on their average predicted co-editing probability exceeding 10% across all businesses in the dataset, ensuring that the final set represented both statistically robust associations and practically meaningful co-editing relationships with sufficient frequency to warrant inclusion in the dynamic weighting framework. This subset was used for further analysis in Section 3.2.3.

3.2.1 Visualization and Assessment of Pairwise Relationships

Significant co-editing patterns were analyzed visually as well as quantitatively. The system of relationships was modeled as a network graph (Figure 3.2) [3], where nodes represent the top 20 variables, and an edge connects two nodes if their co-editing relationship is significant. Edge thickness was weighted by the χ^2 statistic, indicating association strength. The network visualization demonstrates the potential of the Generalized Fellegi-Holt approach, showing that conceptually related variables (such as different person-count variables) are frequently edited together, demonstrating that error localization benefits from considering dependent variable relationships, as discussed in Chapter 2.2.

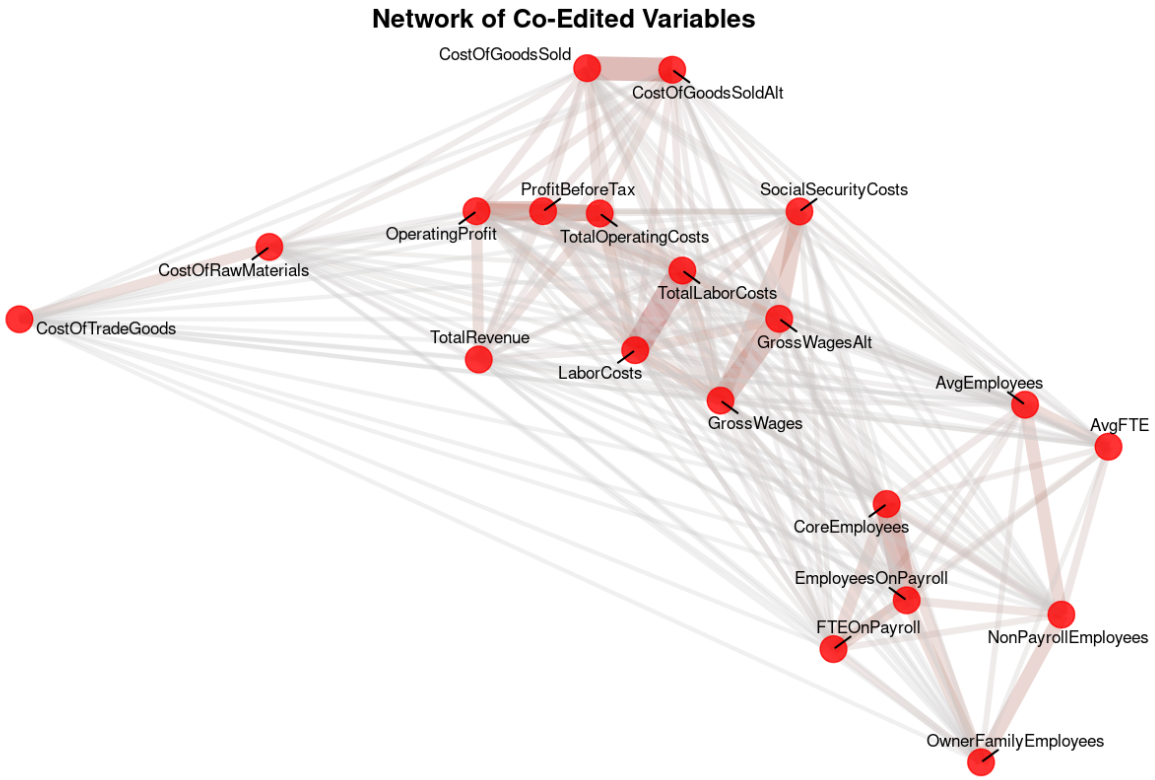


Figure 3.2: Network Analysis Plot for Significant Pairs (χ^2)

The nature of dependencies was explored by assessing directionality using conditional probabilities for each significant pair (v_i, v_j):

$$P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1) \quad \text{and} \quad P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1) \tag{3.3}$$

To quantify the asymmetry between these probabilities, the following metrics were computed

[3]Due to confidentiality practices and for further enhanced understanding, raw variable names were converted into anonimized descriptive variable names. Further information regarding variables from this survey and their meaning can be found in Dutch through CBS microdata. (Statistics Netherlands (CBS), 2024)

for each pair:

Probability Difference:

$$\Delta P_{ij} = |P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1) - P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1)| \tag{3.4}$$

Probability Ratio:

$$R_{ij} = \frac{\max(P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1), P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1))}{\min(P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1), P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1))} \tag{3.5}$$

Relationship Strength:

$$S_{ij} = \frac{P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1) + P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1)}{2} \tag{3.6}$$

Using these metrics, each significant pair was classified according to the following formal criteria:

Mutual Relationship: The relationship is considered mutual, indicating a symmetric co-editing pattern, if there is little to no asymmetry. This is true if the absolute probability difference is small ($\Delta P_{ij} < 0.1$) or the probability ratio is close to one ($R_{ij} < 1.2$). This category is further subdivided into "Exact Mutual" for cases of near-perfect symmetry ($\Delta P_{ij} < 0.01$).

Strong Directional Relationship: The relationship is strongly directional, indicating a clear asymmetric dependency, if the probability ratio is large ($R_{ij} > 1.4$). For these relationships, the direction of dependency is explicitly determined by identifying which conditional probability is greater:

$$\text{Direction}_{ij} = \begin{cases} v_j \rightarrow v_i & \text{if } P(v_i^{\text{flag}} = 1 \mid v_j^{\text{flag}} = 1) > P(v_j^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1) \\ v_i \rightarrow v_j & \text{otherwise} \end{cases} \tag{3.7}$$

Weak Directional Relationship: Any remaining case that exhibits moderate asymmetry (i.e., where $\Delta P_{ij} \geq 0.1$ and $1.2 \leq R_{ij} \leq 1.4$) is classified as weakly directional.

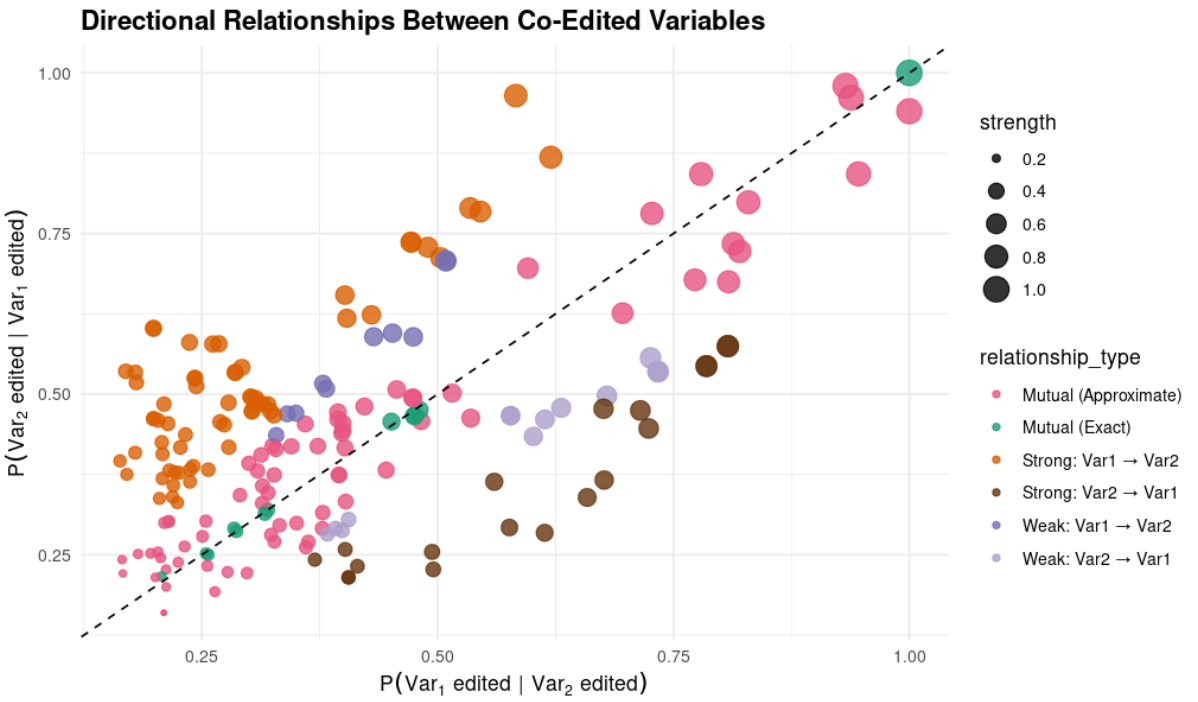


Figure 3.3: Pairwise Directionality Plot

The directionality plot shows that many co-editing relationships are asymmetric, with strong directional dependencies accounting for a significant portion of the pairs. Mutual relationships, represented by points near the diagonal, are less common. It can be noted that the classical Fellegi-Holt independence assumption is often violated. Furthermore, future error localization algorithms could benefit from incorporating directional constraints to better reflect observed hierarchical correction patterns.

3.2.2 Stratified Analysis by Business Characteristics

This analysis examined whether error probabilities vary systematically across different business types to validate using business characteristics as predictors in logistic regression models. The edit-flag matrix was merged with business demographic data including industry sector (2-digit NACE [4] codes), pre-defined size classification (based on the number of employees in a company into Small, Medium, Large), legal structure (natural person vs. legal entity), structural complexity (single-unit vs. multi-unit businesses), and establishment timing [5] relative to the survey year.

[4]NACE (Nomenclature of Economic Activities) codes are the European standard for classifying business activities by industry sector. The 2-digit level provides broad industry categories (e.g., "01" for crop and animal production, "47" for retail trade).

[5]The establishment timing records the month when the business began operations. Businesses were classified as *new* if their establishment date fell within the survey reference year, versus *established* for businesses that began operations before the survey year.

One-way ANOVA tested for differences in edit rates across industry sectors, focusing on NACE codes with at least 15 observations. Two-sample t-tests compared edit rates between legal structures, while Wilcoxon rank-sum tests examined differences across size categories.

The analysis revealed substantial heterogeneity in editing patterns across all examined characteristics. ANOVA results showed significant variation in edit rates across industry sectors for most variables tested. T-tests identified significant differences between natural persons and legal entities, with natural persons showing consistently higher edit rates for labor-related variables. Size-based comparisons revealed that medium and large enterprises had systematically different error patterns compared to small businesses. Multi-unit businesses exhibited different editing patterns than single-unit operations, and businesses established within the survey year showed distinct error profiles compared to those established before the survey period.

This heterogeneity across all business characteristics validated their use as predictors in the subsequent logistic regression models. The observed variation confirms that business context influences error probability, supporting the use of dynamic weighting schemes that adjust correction probabilities based on observable business characteristics. Detailed results including industry-specific edit rate heatmaps are provided in the Appendix.

3.2.3 Predictive Modeling for Reliability Weights

This stage transitions from analyzing past editing to developing a predictive framework for estimating error probabilities based on record characteristics.

Logistic regression was selected for its suitability for binary classification and its interpretable coefficients. The framework models the probability of an error event by expressing its log-odds as a linear combination of specific business characteristics. The general form:

$$\log\left(\frac{p_{\text{event}}}{1-p_{\text{event}}}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m \tag{3.8}$$

is specified for each business record k using the following predictors:

$$\log\left(\frac{p_{\text{event},k}}{1-p_{\text{event},k}}\right) = \beta_0 + \beta_{\text{size}} x_{\text{size},k} + \beta_{\text{legal}} x_{\text{legal},k} + \beta_{\text{units}} x_{\text{units},k} + \beta_{\text{new}} x_{\text{new},k} \tag{3.9}$$

where $p_{\text{event},k}$ is the probability of an error event for record k, and the predictors are:

- $x_{\textbf{size},k}$: The business size category (Small, Medium, or Large).
- $x_{\textbf{legal},k}$: A binary indicator for the legal structure (natural person vs. legal entity).
- $x_{\textbf{units},k}$: A binary indicator for structural complexity (multi-unit vs. single-unit).

- $x_{\mathbf{new},k}$: A binary indicator identifying businesses established within the survey year.

The terms β_j are the estimated coefficients representing the change in log-odds associated with each predictor level, relative to a defined reference category.

Two sets of logistic regression models were constructed:

1. **Individual Variable Models:** For each of the most frequently edited variables, a logistic regression model predicts the probability of manual correction. The response variable is the binary edit flag v_{ik}^{flag} for variable v_i in record k, where $v_{ik}^{\text{flag}} = 1$ if the variable was manually corrected and 0 otherwise.

2. **Pairwise Co-editing Models:** For each statistically significant variable pair chosen earlier(v_i, v_j), a logistic regression model predicts joint editing probability. The response variable is defined as $v_{ik}^{\text{flag}} \times v_{jk}^{\text{flag}}$, where the result equals 1 if both variables i and j were simultaneously edited in record k, and 0 otherwise. These models inform the weighting of generalized edit operations (w_k^{EO} in Equation 2.4) targeting specific co-editing patterns.

Empirical Static Reliability Weights

Each fitted logistic regression model generates predicted probabilities that are transformed into reliability weights for the GFH optimization framework. For individual variable models, the predicted probability $\hat{p}_i$ represents the average likelihood across all records that variable v_i requires correction. The probability of correctness is calculated as $\hat{q}_i = 1 - \hat{p}_i$.

As per equation 2.13 reliability weights are derived as:

$$w_i = \log\left(\frac{\hat{q}_i}{1-\hat{q}_i}\right) = \log\left(\frac{1-\hat{p}_i}{\hat{p}_i}\right) \tag{3.10}$$

This process generates static weights: one weight per variable for individual models and one weight per variable pair for pairwise models.

The resulting weights are rounded to five decimal places and exported in semicolon-delimited format to be used in the error localization algorithm discussed in Chapter 4.

In subsequent sections, this framework will be extended to generate record-specific weights w_{ik} that adapt dynamically to individual business contexts, enabling more precise error localization tailored to specific enterprise characteristics.

3.2.4 Record-Specific Dynamic Reliability Weights

While the static weights provide an initial baseline, the analysis in section 3.2.2 confirmed that error probabilities are not uniform across different business subpopulations. This finding motivates a more advanced approach: a dynamic weighting scheme where reliability weights are constructed for each individual business record.

This version of the model utilizes the same fitted logistic regression models, but applies them to generate a unique predicted probability on a record-by-record basis - $\hat{p}_{\text{event},k}$, for each potential error event (individual or co-editing) and for each specific record k. This is achieved by inputting the specific predictor values ($x_{\text{size},k}, x_{\text{legal},k}$, etc.) for record k into the relevant fitted logistic model.

The resulting record-specific probability of error for an individual variable i is denoted $\hat{p}_{ik}$, and for a co-editing pattern m is denoted $\hat{p}_{mk}$. The corresponding probability of correctness for record k is $\hat{q}_{ik} = 1 - \hat{p}_{ik}$ and $\hat{q}_{mk} = 1 - \hat{p}_{mk}$.

The dynamic reliability weight for individual variable i and record k, denoted w_{ik}, is then calculated using the logit transformation:

$$w_{ik} = \log\left(\frac{\hat{q}_{ik}}{1 - \hat{q}_{ik}}\right) = \log\left(\frac{1 - \hat{p}_{ik}}{\hat{p}_{ik}}\right) \tag{3.11}$$

Similarly, the dynamic weight for generalized edit operation m and record k, denoted w_{mk}, is:

$$w_{mk} = \log\left(\frac{1 - \hat{p}_{mk}}{\hat{p}_{mk}}\right) \tag{3.12}$$

The output of this process is not a single vector of weights, but a mechanism to generate a unique weight vector for each incoming data record. This allows the reliability of each potential correction to be assessed based on the specific context of the business in question.

Table 3.2: Illustrative example of static reliability weights for variables A, B, and C.

Business ID	**Individual Weights** (w_i)			**Pairwise Weights** (w_m)	
	A	B	C	pair_AB	pair_AC
1012	2.135	3.109	4.155	1.386	5.991
1032	2.135	3.109	4.155	1.386	5.991
1035	2.135	3.109	4.155	1.386	5.991

The practical application of both the static and dynamic weighting schemes, including how they are integrated as inputs into the error localization solver, will be detailed in the subsequent chapter (Chapter 4) on implementation.

Table 3.3: Illustrative example of dynamic (record-specific) reliability weights for variables A, B, and C.

Business ID	**Individual Weights (w_{ik})**			**Pairwise Weights (w_{mk})**	
	A	B	C	pair_AB	pair_AC
1012	2.581	1.987	4.050	0.875	6.132
1032	1.766	3.210	4.312	1.450	4.889
1035	2.054	3.001	5.231	1.303	5.950

3.3 Higher-Order Association Analysis

The pairwise association analysis detailed in Section 3.2 identifies variables that are frequently co-edited in pairs. However, it is plausible that more complex error mechanisms exist, involving the simultaneous correction of three, four, or even more variables. Such higher-order relationships (triplets, quadruplets, etc.) may not be fully captured by analyzing pairs in isolation. This section details the methodology used to discover these higher-order co-editing patterns and to assess their statistical significance beyond what can be explained by their constituent pairwise interactions. While the focus here is specifically on triplets and quadruplets, the described methods are generalizable to higher-order cases.

3.3.1 Discovery of Frequent Higher-Order Itemsets using the Eclat Algorithm

Identifying frequently co-edited sets of three or four variables using contingency tables is computationally infeasible due to the number of combinations. Therefore, a frequent itemset mining approach was adopted, utilizing the Eclat algorithm (Zaki et al., 1999) for this task. The Eclat algorithm constructs frequent itemsets by representing each item with a list of all the transactions in which it appears and then recursively intersecting these lists.

In this application, the "items" are the survey variables that were edited, and a "transaction" corresponds to a single business record represented by a row in the binary edit-flag matrix. The Eclat algorithm was applied to a subset of this matrix, including all variables that were edited with sufficient frequency to be of interest, similar to before. The objective was to identify all sets of variables (itemsets) of size three (triplets) and four (quadruplets) that were co-edited in a number of records exceeding a specified minimum support threshold. The support of an itemset (e.g., $\{v_a, v_b, v_c\}$) is defined as the proportion of records in which all items in the set were co-edited:

$$\text{Support}(\{v_a, v_b, v_c\}) = \frac{\text{Number of records where } v_{ak}^{\text{flag}} \cdot v_{bk}^{\text{flag}} \cdot v_{ck}^{\text{flag}} = 1}{\text{Total number of records}} \tag{3.13}$$

A minimum support threshold of 5% was used to generate a manageable list of candidate higher-order co-editing patterns for further statistical validation.

3.3.2 Testing for true Higher-Order Interaction with Log-Linear Models

The identification of a frequent itemset (e.g., a triplet $\{v_a, v_b, v_c\}$) does not, by itself, confirm a true three-way statistical interaction. The high frequency of the triplet could be a consequence of strong underlying pairwise associations (e.g., strong associations between $\{v_a, v_b\}$, $\{v_a, v_c\}$, and $\{v_b, v_c\}$). To distinguish true higher-order effects from these constituent lower-order effects, log-linear models were employed.

A log-linear model analyzes the cell counts in a multi-way contingency table. To test for a k-way interaction, two nested models are compared. The null model (H_0) includes all interaction terms up to order $k-1$, while the alternative model (H_1, the saturated model for this test) includes all terms in H_0 plus the k-way interaction term. The significance of the k-way interaction is assessed using a likelihood-ratio test, which compares the fit of these two models. The test statistic, often denoted G^2 asymptotically follows a χ^2 distribution under the null hypothesis.

For a triplet of indicator variables (A, B, C), the test compares the fit of a model containing only main effects and two-way interactions against the saturated model:

$$H_0 : \log(E_{ijk}) = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} \tag{3.14}$$

$$H_1 : \log(E_{ijk}) = \log(E_{ijk})_{H_0} + \lambda_{ijk}^{ABC} \tag{3.15}$$

A significant G^2 value suggests the three-way interaction term λ_{ijk}^{ABC} is necessary to adequately explain the observed frequencies. An analogous procedure was applied to the frequent quadruplets identified by Eclat to test for the significance of the four-way interaction term. This process isolates higher-order patterns that represent true statistical interactions.

For a quadruplet of indicator variables (A, B, C, D), the test for a true four-way interaction involves comparing the following nested log-linear models:

The null model (H_0) for this test posits that there is no four-way interaction. It assumes that the observed frequencies in the $2 \times 2 \times 2 \times 2$ contingency table can be fully explained by all lower-order effects, including main effects, all two-way interactions, and all three-way interactions. The alternative model (H_1) is the saturated model, which includes all terms from the null model

plus the four-way interaction term, λ_{ijkl}^{ABCD}.

$$
\begin{aligned}
H_0: \quad \log(E_{ijkl}) = \mu &+ \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_l^D \\
&+ \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{il}^{AD} + \lambda_{jk}^{BC} + \lambda_{jl}^{BD} + \lambda_{kl}^{CD} \\
&+ \lambda_{ijk}^{ABC} + \lambda_{ijl}^{ABD} + \lambda_{ikl}^{ACD} + \lambda_{jkl}^{BCD} \qquad (3.16) \\
H_1: \quad \log(E_{ijkl}) = \log(E_{ijkl})_{H_0} &+ \lambda_{ijkl}^{ABCD} \qquad (3.17)
\end{aligned}
$$

Similarly, a significant G^2 value from the likelihood-ratio test comparing these two models indicates that the four-way interaction term λ_{ijkl}^{ABCD} is necessary to explain the observed frequencies, confirming a true quadruplet association.

3.3.3 Hierarchical Absorption Analysis and Non-Redundant Operation Definition

The identification of statistically significant pairs, triplets, and quadruplets results in a large, overlapping set of potential generalized edit operations. For instance, a significant triplet pattern $\{v_a, v_b, v_c\}$ inherently contains three pairwise patterns. Defining separate edit operations for both the triplet and its constituent pairs would be redundant and could lead to incorrect weight assignments in the GFH framework.

To address this, a "hierarchical absorption" analysis was conducted. The principle is that a lower-order pattern is considered "absorbed" by a higher-order pattern if its occurrence is largely explained by the occurrence of that higher-order pattern. The goal is to create a set of non-redundant edit operations where each error phenomenon is represented by its most specific, highest-order pattern. The absorption of a pair $\{v_i, v_j\}$ by a triplet $\{v_i, v_j, v_k\}$ is quantified by the conditional probability:

$$
A(\{v_i, v_j\} \leftarrow \{v_i, v_j, v_k\}) = \frac{\text{Count}(\{v_i, v_j, v_k\})}{\text{Count}(\{v_i, v_j\})} = P(v_k^{\text{flag}} = 1 \mid v_i^{\text{flag}} = 1 \wedge v_j^{\text{flag}} = 1) \quad (3.18)
$$

The maximum absorption of a given pair by any significant triplet it is part of was calculated. A similar calculation was performed for the absorption of pairs by quadruplets, and of triplets by quadruplets.

Based on this analysis, a final set of edit operations was selected using empirically determined thresholds. For instance, a significant pair was chosen as a final edit operation only if it was not significantly absorbed by any higher-order triplet or quadruplet (in this case threshold absorption ratios were empirically set to 0.75). This procedure yields a refined, non-redundant hierarchy of edit operations (pairs, triplets, and quadruplets) representing distinct error mechanisms.

3.3.4 Hierarchical Dynamic Reliability Weights

The final step of the methodology integrates the predictive models with the absorption analysis to generate a single, coherent set of dynamic reliability weights for use in the error localization algorithm. This approach adjusts the probability of a lower-order error based on the extent to which it is "absorbed" by more complex, higher-order patterns.

$$\hat{q}_{ik}^{\text{final}} \approx (1 - A_{i \leftarrow \text{pairs}}) \times (1 - \hat{p}_{ik}^{\text{base}}) \tag{3.19}$$

where $A_{i \leftarrow \text{pairs}}$ is the maximum absorption ratio for variable v_i by any significant pair.

Similarly, for a co-editing pair $\{v_i, v_j\}$, its probability of occurring as a unique pair (and not as part of a triplet) is adjusted:

$$\hat{q}_{mk}^{\text{final}} \approx (1 - A_{\{i,j\} \leftarrow \text{triplets}}) \times (1 - \hat{p}_{mk}^{\text{base}}) \tag{3.20}$$

where $A_{\{i,j\} \leftarrow \text{triplets}}$ is the maximum absorption ratio for the pair by any significant triplet.

These final, adjusted probabilities of correctness, $\hat{q}^{\text{final}}$, are then transformed into the final dynamic reliability weights using the standard logit transformation (Equation 3.10). This hierarchical adjustment ensures that when the MIP solver considers corrections for a record, the weight for a comprehensive, higher-order operation (e.g., a triplet) is appropriately lower (more favorable) than the combined weights of its constituent, less-complete parts (e.g., the pairs within it), guiding the solver to the most parsimonious and contextually accurate explanation for the observed errors. The implementation of this complete dynamic and hierarchical weighting scheme is detailed in the following chapter.

Chapter 4

Implementation and Evaluation Framework

This chapter details the practical implementation of the methodologies developed in Chapter 3 and establishes the framework for their evaluation. The primary objective is to translate the theoretical model for discovering error patterns and deriving empirical reliability weights into a functional system. The implementation was conducted entirely within the R statistical programming environment, leveraging the `errorlocate` (van der Loo and de Jonge, 2018) package for solving the Generalized Fellegi-Holt problem.

The central hypothesis is that empirically derived, record-specific reliability weights can improve the performance of automated error localization compared to predefined and static approaches. To test this hypothesis, six distinct versions of the GFH weighting scheme were implemented in an incremental progression, ranging from a baseline using predefined weights to a hierarchical system that incorporates higher-order error patterns and record-specific characteristics.

This chapter first describes the technical architecture and workflow of the error localization framework. It then details the operationalization of each weighting version, explaining how the theoretical models from Chapter 3 were translated into inputs for the GFH solver. Finally, it outlines the evaluation methodology, including performance metrics and comparison strategies that will be employed in Chapter 5.

4.1 Technical Implementation of Error Localization

The GFH error localization problem was solved using the `errorlocate` package in R. The implementation consists of a parallelized workflow that processes the entire dataset efficiently by dividing it into manageable chunks distributed across multiple CPU cores. This architecture was

necessary to handle the computational demands of processing approximately 16,000 business records, each containing around 100 variables, across multiple weighting schemes.

The system follows a modular two-phase architecture designed for scalability and maintainability. The first phase handles data preparation and weight generation, while the second phase executes the error localization algorithm. This separation allows for independent testing and validation of different weighting schemes without modifying the core solver logic [1].

4.1.1 Data Processing and Preparation

The initial data processing phase transforms raw survey data into the format required by the error localization algorithm. Business records are loaded with their associated metadata including size classification, legal structure, operational complexity, and establishment timing. These demographic characteristics serve as predictor variables for the dynamic weighting schemes.

The edit-flag matrix construction, as described in Section 3.1, matches each business record from the raw dataset with its manually corrected counterpart, creating binary indicators that identify variables requiring expert intervention and mark the locations where manual corrections were applied.

Validation rules are loaded from a master rule repository and filtered based on questionnaire type and variable availability. This filtering ensures that each rule applies only to records containing all referenced variables, preventing solver failures due to undefined variable references. The rule filtering process adapts automatically to different questionnaire versions used across survey years.

4.1.2 System Architecture and Workflow

In the setup phase, each business record is prepared and the reliability weights vector is constructed, serving as the primary injection point where different weighting schemes are operationalized.

The weight vector construction process varies significantly depending on the weighting approach. Static approaches use pre-calculated weight files applied uniformly to all records, while dynamic approaches load fitted logistic regression models and generate weights on-demand based on each record's specific characteristics. This dynamic generation requires careful memory management and model caching to maintain computational efficiency.

In the execution phase, the prepared record, validation rules, and weights are processed by the

[1] More details on the implementation of the error localization algorithm through `errorlocate` can be found in van der Loo and de Jonge (2018).

error localization algorithm, which formulates the problem as a mathematical model and solves it using mixed-integer programming. The solver configuration includes time limits and quality tolerances to balance solution accuracy with computational efficiency.

The solver seeks the optimal solution that minimizes the total weight of applied corrections while ensuring the final record satisfies all validation rules. The output identifies which variables are deemed erroneous and which generalized edit operations should be applied. Results are aggregated and exported into structured CSV files including error identification results, reliability weight matrices, rule confrontation outcomes, and processing status logs. Status codes track solver outcomes including 0 (optimal solution found), 1 (possibly sub-optimal solution), and *other* values for various computational states such as infeasibility, timeouts, and numerical failures.

The key advancements are the systematic, empirical derivation of reliability weights and the implementation of hierarchical absorption adjustments for complex error patterns. Instead of applying identical weights to all records, the system generates unique weights for each record based on specific business characteristics. The hierarchical approach further refines these weights by accounting for the overlap between different error patterns, ensuring that higher-order patterns take precedence over their constituent lower-order components. This approach required modifications to the standard error localization process, including new weight calculation procedures, defining edit operations, and integration with the fitted regression models.

4.2 Development of Weighting Schemes

The implementation strategy follows a controlled experimental design where each version builds upon previous versions by adding specific methodological components. This incremental approach enables precise identification of the performance impact of individual innovations, such as the transition from static to dynamic weights or the incorporation of higher-order error patterns.

A critical component of the implementation is the definition of generalized edit operations that enable the error localization algorithm to consider simultaneous corrections to multiple variables.

Based on the empirical analysis from Chapter 3, the top 50 most frequently co-edited variable pairs (as seen in Section 3.2) were identified and implemented as generalized edit operations for all versions. Each pairwise edit operation is defined through a structured framework that specifies the target variables and their correction relationships. For example, the operation for two personnel variables (P_1 - total personnel count in one department and P_2 - total personnel count in another) allows the algorithm to simultaneously adjust both, recognizing that these

variables are conceptually related and often require coordinated corrections.

For higher-order relationships, the implementation extends beyond pairwise operations. The frequent triplets and quadruplets identified through the Eclat algorithm and validated with log-linear models are implemented as additional generalized edit operations. Similar to pairwise edit operations, these operations enable corrections to variable triplets or quadruplets, with each higher-order operation having its own auxiliary variables that the solver considers when selecting the most cost-effective correction approach.

4.2.1 Version Implementation

Seven versions were implemented to evaluate different weighting strategies, progressing from static baseline approaches to dynamic, hierarchical systems. The versions are organized into static weight schemes (Versions 1-3) that use fixed weights across all records, and dynamic weight schemes (Versions 4-7) that generate record-specific weights.

Individual variable weights are derived from logistic regression models that predict correction probability based on business characteristics including size, legal structure, operational complexity, and establishment timing. Pairwise weights are calculated by converting predicted co-editing probabilities into reliability weights using the log-odds transformation from Equation 3.10. Higher-order patterns (triplets and quadruplets) are implemented using the same log-odds transformation applied to their respective predicted probabilities with the final version including hierarchical absorption adjustments as defined in 3.3.4 to account for pattern overlap. The key distinctions between versions are whether weights remain fixed across all records or are calculated individually for each record, and whether hierarchical absorption adjustments are applied to improve weight accuracy. Below, details can be found about each.

Original Static Individual Weights (Version 1)

This version, already implemented by CBS, serves as the baseline for comparison, using a set of static reliability weights for 26 individual variables. These weights were not derived empirically from historical data; instead, they were assigned by human experts based on their subject matter expertise and are applied uniformly across all records.

Original Static Individual and Empirical Static Pairwise Weights (Version 2)

Building on the baseline, this version introduces the first layer of empirical information by combining the static expert-defined individual weights with static weights for pairwise co-editing patterns derived from historical data. The pairwise weights have been defined for the 50

most significant pairs, as stated in Section 4.2.

Empirical Static Individual and Empirical Static Pairwise Weights (Version 3)

This version represents a fully empirical, though static, approach. In contrast to Version 2, the predefined individual weights are replaced with weights that are also derived empirically from historical data, while the pairwise weights remain the same. The purpose of this version is to directly compare the performance of expert-defined individual weights against fully empirical ones, while keeping the pairwise weights the same.

Original Static Individual and Dynamic Pairwise Weights (Version 4)

This version introduces record-specific weight generation for pairwise co-editing patterns while keeping individual variable weights predefined and static. The hybrid approach enables direct measurement of the impact of dynamic adaptation for complex errors. For each business record and each significant pairwise operation, the corresponding logistic regression model generates a record-specific predicted probability of co-editing. These probabilities are converted into dynamic weights using the log-odds transformation, resulting in a unique weight vector for every record, as seen in Table 3.3.

Dynamic Individual and Dynamic Pairwise Weights (Version 5)

This version extends the dynamic approach to all reliability weights, creating a fully adaptive system, where both individual and pairwise weights adjust to record-specific characteristics. This represents the direct operationalization of the predictive modeling framework from Section 3.2.3.

Both individual and pairwise logistic regression models generate record-specific error probabilities. The system applies these models sequentially to generate the complete dynamic weight vector for each record.

Dynamic Individual and Dynamic Higher Order Weights (Version 6)

This version extends the dynamic approach to include higher-order error patterns. The most significant triplets and quadruplets were selected for inclusion based on the G^2 goodness-of-fit test statistic (as discussed in Section 3.3.2). The threshold was determined empirically and set at the 0.7 quantile (to include the top 30% of most significant triplets and quadruplets) of the test statistic's distribution to balance the number of defined edit operations against model

complexity. For each of these selected 56 triplets and 17 quadruplets, edit operations were defined, similarly as to how they were defined for pairs. This version generates record-specific weights for all pattern levels (individual, pairwise, triplet, and quadruplet).

Hierarchical Dynamic Weights for All Pattern Orders (Version 7)

This version represents the complete implementation of the full methodology from Chapter 3. It incorporates dynamic weights for all pattern orders (individual, pairs, triplets, and quadruplets) and includes hierarchical absorption adjustments to manage relationships between patterns.

The implementation calculates the base dynamic probabilities for all potential errors for each record, as was done in Version 6. It then applies the pre-calculated absorption ratios from Section 3.3.3 to adjust these probabilities, as described in Equation 3.19. In total, 56 pairs, 111 triplets, and 17 quadruplets [2] were included. This adjustment accounts for statistical dependence by reducing the probability assigned to a lower-order pattern, effectively attributing a portion of its occurrence to the more comprehensive, higher-order pattern that contains it. Overall, the final adjusted probabilities are converted into the set of weights passed to the solver.

The goal of this version is to assess whether incorporating absorption improves performance compared to the non-hierarchical approach used in Version 6.

4.3 Evaluation Methodology

To assess and compare the performance of each implemented weighting version, a comprehensive evaluation strategy was defined. The primary goal is to measure how accurately each version identifies the true set of variables that were corrected by human experts. The evaluation framework employs standard binary classification metrics while accounting for the specific characteristics of the error localization problem.

The evaluation approach recognizes that error localization performance depends on both the accuracy of individual variable predictions and the effectiveness of generalized edit operations. The multi-faceted comparison strategy enables assessment of performance across different types of errors and correction patterns, providing insights into the strengths and limitations of each weighting approach.

[2] The number of quadruplets doesn't change in comparison to Version 6, as the absorption calculation doesn't affect quadruplets.

4.3.1 Data Integration and Analysis Strategy

The analysis utilizes the complete dataset of manually edited records from both available survey years (2021 and 2022) in an integrated approach. All logistic regression models, weight calculations, and error localization algorithms are applied to this full dataset to maximize statistical power for discovering co-editing patterns, capture comprehensive error patterns across different survey cycles, and enable direct performance assessment through comparison with expert decisions.

This integrated approach treats the manual editing decisions as the reference standard and evaluates how well different weighting schemes can reproduce expert decisions when applied to the same records. The methodology focuses on optimizing the error localization process to better align automated decisions with the patterns observed in manual corrections performed by domain experts.

4.3.2 Performance Metrics and Evaluation Framework

For each record in the test set, the ground truth is the known set of edited fields, as represented by the edit-flag matrix where $v_{ik}^{\text{flag}} = 1$ indicates that variable i in record k was manually corrected by domain experts. Each version of the error localization algorithm was run on the raw version of the test records, producing a predicted set of erroneous fields for each record under each weighting scheme. The predicted set was then compared to the true set from the ground-truth data.

Performance was quantified using standard binary classification metrics, aggregated across all records in the test set. These metrics provide interpretable measures of algorithm effectiveness that can be compared across different versions and contextualized relative to operational requirements. The evaluation treats each variable in each record as an independent classification decision, aggregating results across the entire test set.

The primary evaluation metrics are based on the following counts:

- **True Positives (TP)**: The number of variables correctly identified by the algorithm as being erroneous (i.e., the algorithm flagged the variable and the expert also edited it).
- **False Positives (FP)**: The number of variables incorrectly identified as erroneous (i.e., the algorithm flagged the variable but the expert did not edit it).
- **False Negatives (FN)**: The number of variables that were actually edited by experts but were missed by the algorithm (i.e., the expert edited the variable but the algorithm did not flag it).

From these counts, the primary evaluation metrics are calculated:

- **Precision**: The proportion of identified errors that were correct ($\frac{TP}{TP+FP}$). This measures the accuracy of the algorithm's positive predictions and indicates the reliability of the algorithm's error identifications from an operational perspective.
- **Recall (Sensitivity)**: The proportion of actual errors that were identified by the algorithm ($\frac{TP}{TP+FN}$). This measures the completeness of the algorithm and indicates how effectively it identifies errors that require correction.
- **F1 Score**: The harmonic mean of precision and recall ($2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$), providing a single, balanced metric that is useful when there is a trade-off between precision and recall.

Although the new versions (2-7) employ pairwise and higher-order edit operations that simultaneously target multiple variables, the evaluation framework remains consistent across all versions because these operations can be deconstructed easily into individual variable corrections. Each pairwise or higher-order edit operation simply identifies multiple variables as requiring correction without imposing specific constraints on the type of correction needed. This allows direct translation back to individual variable-level binary decisions (edited/not edited) for evaluation purposes. This approach differs from more complex edit operations such as value swaps or conditional corrections, which would require specialized evaluation frameworks to properly assess their effectiveness.

4.3.3 Comparison Types

To provide a comprehensive view of performance across different types of errors and correction patterns, these metrics were calculated for three distinct comparison subsets of variables:

1. **Full Comparison**: Compares the predicted and true errors across all variables in the dataset. This provides an overall assessment of algorithm performance across the complete range of survey variables and error types.
2. **Focused on Pairwise Operations**: Compares performance only on the subset of variables that are part of the defined pairwise edit operations identified in Section 3.2. This focused evaluation assesses how well the models perform on the specific types of co-editing errors they were designed to recognize and correct.
3. **Focused on Higher-Order Operations**: Compares performance only on the subset of variables involved in the higher-order patterns (triplets and quadruplets) identified in

Version 6 through the analysis in Section 3.3. This evaluation targets complex multi-variable error patterns and assesses the effectiveness of higher-order dynamic weighting.

4. **Focused on Hierarchical Operations**: Compares performance only on the subset of variables involved in the complete hierarchical framework implemented in Version 7, including all individual, pairwise, and higher-order patterns with absorption adjustments. This evaluation assesses the effectiveness of the full hierarchical weighting approach with absorption mechanisms.

This comparison strategy enables identification of version-specific strengths and weaknesses, revealing whether improvements are concentrated in particular types of errors or distributed across the full range of correction patterns. The focused comparisons provide insights into the effectiveness of the empirical pattern discovery methodology and the value of targeting specific error types with tailored approaches.

The results of this evaluation across all versions and comparison types are presented and analyzed in the following chapter, providing empirical evidence for the effectiveness of the proposed data-driven approaches to weight generation.

Chapter 5

Results

This chapter presents the empirical evaluation of the weighting schemes developed in Chapter 3 and implemented in Chapter 4. Each version of the error localization system was applied to the survey dataset mentioned in Section 3.1. The objective of the evaluation is to quantify the improvement in error detection performance across the weighting versions and to assess the operational feasibility of each approach in terms of computational efficiency and solver behavior.

The results are organized into three sections. First, the performance of each version is compared across four different comparison subsets (full dataset, pairwise-focused, higher-order focused, and hierarchical-focused as seen in 4.3.3), using standard classification metrics to evaluate how well each automated approach replicates the expert manual corrections. Second, computational performance across versions is reported, including solver convergence rates and median processing time per record. Finally, the findings are summarized to analyze the practical implications for deployment in production survey editing environments.

5.1 Performance Metrics Across Versions

The central hypothesis evaluated in this chapter is that empirically grounded and record-specific reliability weights can improve the alignment between automated error localization and the manual corrections performed by domain experts. To assess this hypothesis, Table 5.1 reports the classification performance metrics - true positives (TP), false positives (FP), false negatives (FN), precision, recall, and F1 scores - for each version across different variable subsets.

Across all comparison types, versions incorporating empirical or dynamic elements consistently outperform the baseline. Version 2, which introduces empirically derived pairwise weights, improves substantially across all evaluations. In the full comparison, precision increases from roughly 0.54 in Version 1 to 0.58 in Version 2, indicating that a higher proportion of automatically

identified errors correspond to actual expert corrections. This precision gain is particularly valuable in the operational context of survey editing, where false positives incur significant costs by potentially overwriting valid business responses or triggering unnecessary manual review processes.

Versions 3 through 5 show progressive performance improvements by incorporating empirical individual weights and record-specific predictions. Version 5, which applies dynamic weights to both individual variables and pairwise operations, achieves the most balanced performance among non-hierarchical systems while maintaining computational efficiency (see Figure 5.1b).

Version 7, implementing the full hierarchical approach, produces the highest absolute number of true positives ($TP = 8,469$), due to its higher capacity to detect complex, multi-variable error patterns (see Figure 5.1a). However, this results in reduced precision due to increased false positives. This characteristic reflects hierarchical generalization: by modeling sophisticated and overlapping error patterns, the system becomes more inclusive in its corrections, achieving broader error coverage at the expense of occasional over-correction.

5.2 Computational Efficiency and Solver Performance

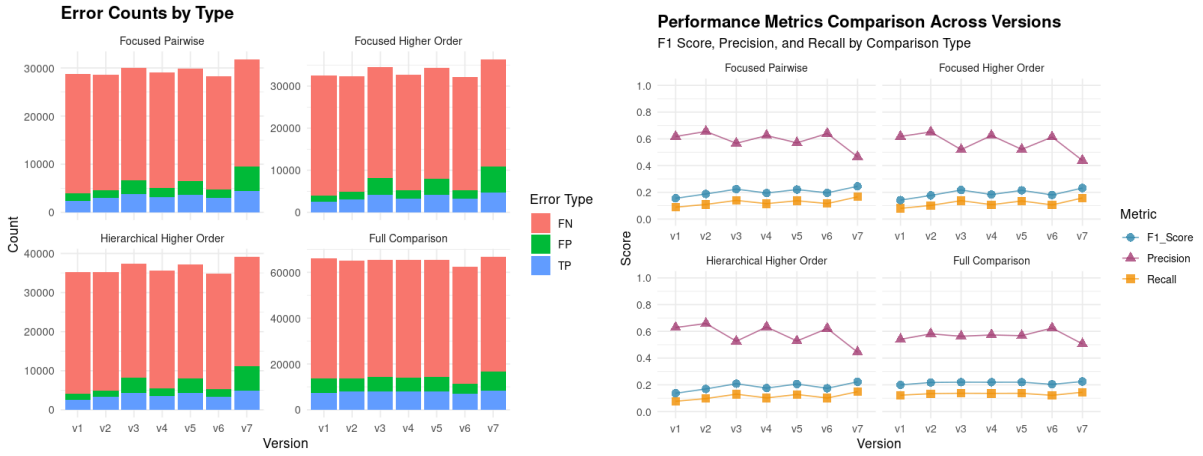
Table 5.3 reports solver performance characteristics by version, including status code distributions and median processing times per record.

Versions 1 through 5 maintain efficient performance profiles, with median processing times below roughly 0.13 seconds per record and high rates of optimal solution convergence ($status = 0$). This efficiency makes these versions suitable for integration into real-time or large-scale editing pipelines. Version 5, despite employing fully dynamic weights, achieves processing times comparable to Version 3, demonstrating that record-specific predictions remain computationally manageable within the current parallelized architecture.

Versions 6 and 7 impose greater computational demands due to higher-order patterns and hierarchical absorption mechanisms. The transition from Version 6 to Version 7 doubles processing time from roughly 0.41 to 0.84 (see Table 5.2) seconds per record, while optimal solution rates decrease to 87.3% (13,963 out of 15,960) for Version 7. These versions exhibit increased frequencies of sub-optimal solutions ($status = 1$) and solver anomalies ($status = other$), indicating increased numerical complexity in the underlying mixed-integer programming formulation. Between the two, in the hierarchical-focused evaluation, the F1 score increases by 28.5% (from roughly 0.18 to 0.23). This gain comes at the expense of precision, which drops from roughly 0.625 to 0.506 in the full comparison.

The hierarchical approach in Version 7 achieves its best performance in the hierarchical-focused

comparison, which aligns with its design objectives for complex pattern recognition. However, Version 6 emerges as potentially more suitable for production environments, offering significant improvements over non-hierarchical approaches while preserving the computational efficiency necessary for large-scale operational deployment.



(a) Error Counts by Type

(b) Performance Metrics Across Versions

Figure 5.1: Visualization of Results Obtained from Error Localization

5.3 Summary of Metrics Evaluation

In summary, the results demonstrate that incorporating empirically derived and record-specific reliability weights improves alignment between automated editing and manual expert corrections. The highest-performing model, Version 7, outperforms alternatives in all subtypes of comparison when it comes to F1 score, achieving improvements of 12.6% in the full comparison (from 0.1996 to 0.2247) and 63.7% in the hierarchical-focused evaluation (from 0.1417 to 0.2319) relative to the baseline. However, these gains come at the cost of increased computational complexity, with processing time increasing from 0.036 seconds per record in Version 1 to 0.84 seconds in Version 7. The next chapter will reflect on these trade-offs, discuss the implications for operational deployment, and outline directions for further research.

Table 5.1: Evaluation Metrics by Version and Comparison Type (Grouped by Version)

Version	Comparison Type	TP	FN	FP	Precision	Recall	F1 Score
v1	Full Comparison	7324	52495	6227	0.5405	0.1224	0.1996
v1	Focused Pairwise	2425	24864	1501	0.6177	0.0889	0.1554
v1	Focused Higher Order	2594	31131	1532	0.6287	0.0769	0.1371
v1	Focused Hierarchical	2475	28445	1528	0.6183	0.0800	0.1417
v2	Full Comparison	7961	51511	5736	0.5812	0.1339	0.2176
v2	Focused Pairwise	2967	24138	1561	0.6553	0.1095	0.1876
v2	Focused Higher Order	3254	30238	1684	0.6590	0.0972	0.1693
v2	Focused Hierarchical	3138	27565	1680	0.6513	0.1022	0.1767
v3	Full Comparison	8106	51165	6287	0.5632	0.1368	0.2201
v3	Focused Pairwise	3773	23283	2886	0.5666	0.1395	0.2238
v3	Focused Higher Order	4352	29084	3920	0.5261	0.1302	0.2087
v3	Focused Hierarchical	4214	26441	3901	0.5193	0.1375	0.2174
v4	Full Comparison	8088	51459	6018	0.5734	0.1358	0.2196
v4	Focused Pairwise	3131	24042	1881	0.6247	0.1152	0.1946
v4	Focused Higher Order	3424	30160	1988	0.6327	0.1020	0.1756
v4	Focused Hierarchical	3306	27484	1985	0.6248	0.1074	0.1833
v5	Full Comparison	8077	51118	6152	0.5676	0.1364	0.2200
v5	Focused Pairwise	3700	23319	2791	0.5700	0.1369	0.2208
v5	Focused Higher Order	4277	29114	3826	0.5278	0.1281	0.2062
v5	Focused Hierarchical	4139	26471	3804	0.5211	0.1352	0.2147
v6	Full Comparison	7087	51229	4252	0.6250	0.1215	0.2035
v6	Focused Pairwise	3073	23493	1728	0.6401	0.1157	0.1959
v6	Focused Higher Order	3323	29514	2024	0.6215	0.1012	0.1741
v6	Focused Hierarchical	3183	26903	2004	0.6136	0.1058	0.1805
v7	Full Comparison	8469	50167	8261	0.5062	0.1444	0.2247
v7	Focused Pairwise	4446	22294	5103	0.4656	0.1663	**0.2450**
v7	Focused Higher Order	4911	28140	6145	0.4442	0.1486	0.2227
v7	Focused Hierarchical	4777	25525	6117	0.4385	0.1576	0.2319

Note: Version definitions —
v1: Original static individual weights; v2: Original static individual and empirical static pairwise weights; v3: Empirical static individual and empirical static pairwise weights; v4: Original static individual and dynamic pairwise weights; v5: Dynamic individual and dynamic pairwise weights; v6: Dynamic individual, and dynamic higher order (pairs, triplets, quadruplets) weights; v7: Hierarchical dynamic weights for all individual variables, pairs, triplets, and quadruplets.

Version	Median Duration
v1	0.0360
v2	0.1171
v3	0.1230
v4	0.1238
v5	0.1131
v6	0.4117
v7	0.8441

Table 5.2: Median Duration of Convergence by Version

Version	0	1	Other
v1	15818	68	74
v2	15508	301	151
v3	15445	316	199
v4	15506	314	140
v5	15450	330	180
v6	14545	947	468
v7	13963	1675	322

Table 5.3: Frequencies of Solution Statuses by Version

Chapter 6

Limitations and Directions for Further Research

This thesis has demonstrated how error probabilities can be estimated directly from historical correction data, and that these estimates can be used to generate empirically grounded reliability weights within the Generalized Fellegi-Holt (GFH) framework. While the results in Chapter 5 demonstrate clear improvements in performance, particularly for higher-order, dynamic weighting schemes - the current methodology is still constrained by a number of practical, methodological, and computational factors. This chapter reflects on these constraints and outlines directions for future work that would support the development of more flexible, adaptive, and operationally viable systems for automatic editing in official statistics.

The methods in this thesis are built on a data-centric learning approach: observed manual corrections are used as signals from which common patterns are extracted, and predictive models are trained to estimate the likelihood of future corrections. However, this approach inherits the characteristics, and the limitations of its input data. Manual corrections may vary over time, between editors, or as a result of evolving validation procedures. This introduces noise into the learning signal. Furthermore, because these corrections are only applied to records already selected for manual review, the models are trained on a non-random subset of the population. The resulting predictions may not generalize well to records that would not typically be reviewed under current workflows, potentially limiting their applicability beyond the edited cohort.

Another limitation is that the data was drawn from only two survey years (2021 and 2022), as described in Section 3.1. While these years provided a sufficient foundation for proof-of-concept modeling, this restricts the models' ability to capture longitudinal patterns or time-dependent shifts in error behavior. As more historical data becomes available in future iterations of this work, the training set can be expanded both temporally and structurally, increasing the statistical strength of the predictions and supporting more robust inferences across different periods.

In terms of predictive features, the current models rely primarily on basic business characteristics—industry classification, legal form, business size, and certain timing indicators. These features proved effective for generating meaningful variable-level and pattern-level weights (as developed in Chapter 3), but they provide only a partial view of the broader business context. The models do not incorporate temporal comparisons (e.g., deviation from a firm's past responses), nor peer comparisons within sectors, which are often informative for detecting anomalies (de Waal et al., 2011; van der Loo and de Jonge, 2018). Additionally, auxiliary data sources, such as VAT returns or other administrative signals, could offer further contextual validation. Expanding the feature space to include such dimensions would likely lead to more accurate and context-aware probability estimates.

From a modeling perspective, logistic regression was chosen for its interpretability, transparency, and numerical stability - factors that are particularly relevant within official statistics. This modeling choice, however, has known limitations. Logistic regression assumes a linear relationship between predictors and the log-odds of an outcome and is less suited to capturing complex, non-linear interactions between variables. As the analysis scales to higher-order patterns like the triplets and quadruplets identified in Section 3.3, these limitations become more apparent. This trade-off between the transparency of traditional methods and the predictive power of more complex algorithms is a central challenge for the modernization of statistical production, as outlined in the broader overview by Beck et al. (2018). Their work highlights that while advanced machine learning models can improve accuracy, their adoption requires careful consideration of transparency and reproducibility.

The most advanced system developed in this thesis (Version 7) demonstrated the strongest empirical performance, due to its use of record-specific weights for individual variables, pairs, and higher-order patterns. However, this version also introduced additional computational complexity. Dynamic weights are generated per record by evaluating a large number of stored prediction models, each representing a generalized edit operation. This increases processing time and makes integration into large-scale production pipelines more challenging. Moreover, because the current system does not incorporate any automated monitoring or retraining process, there is a risk of model degradation over time if data characteristics or editing practices shift. Without routine retraining or adaptation, the accuracy of the weight predictions may decline, particularly in dynamic business environments.

To move from a working prototype to a scalable and adaptive solution, future work should focus on three core areas. First, the quality of the learning signal used for training the predictive models could be improved. The current methodology learns from individual editing events, where each manual correction on a single record is treated as an independent observation. This approach can be sensitive to noise from rare or inconsistent corrections. A more robust methodology would involve first constructing aggregate correction profiles. This means analyzing editing patterns

across thousands of records, potentially over several years, to identify stable, high-frequency error types. For instance, instead of learning from one-off corrections, the model would learn from an aggregate profile that indicates a certain error pattern occurs consistently in 15% of records. Furthermore, these profiles could be made context-specific, for example by creating separate profiles for different business characteristic groups, such as NACE. This would allow the system to learn that certain error types are common in one industry but rare in another, making the resulting reliability weights more targeted and accurate.

Second, the predictive models could be enhanced by incorporating richer contextual features - such as longitudinal behavior, sectoral benchmarks, or auxiliary administrative data. The use of administrative registers to validate and enrich survey data is a well-established methodology for improving data quality, with Wallgren and Wallgren (2014) providing a comprehensive guide to the statistical methods involved. Third, future implementations could include adaptive components, such as automated performance tracking and retraining pipelines, to ensure the system evolves with changing data environments and remains aligned with operational standards.

Another promising direction lies in the incorporation of uncertainty into the predictions. Currently, the system outputs point estimates for each error probability, which are then transformed into reliability weights. However, these estimates do not reflect the model's confidence in its predictions. Introducing uncertainty-aware modeling - such as Bayesian logistic regression or model ensembling - would allow the system to identify records where prediction confidence is low. Such records could be selectively routed for manual review, creating a more efficient workflow aligned with principles of selective editing (de Waal et al., 2011).

Finally, while this thesis focuses on defining and predicting error events at the variable and pattern level, future extensions could explore decision-making strategies that consider the full record holistically. For example, methods based on sequential modeling or policy learning could simulate the behavior of expert editors, who often apply edits iteratively and evaluate the overall plausibility of the record in context. While technically more complex, such approaches would move the system closer to operational practice and offer a more integrated alternative to evaluating generalized edit operations in isolation.

In conclusion, the methodology developed in this thesis provides a functional and empirically grounded framework for data-driven reliability weighting in statistical editing. It also opens up a broader research agenda - one that involves improving learning signals, expanding feature spaces, introducing uncertainty measures, and building adaptive system architectures. Together, these extensions would bring the GFH framework closer to a dynamic, intelligent editing system that learns from real-world corrections, adapts to new data, and supports evolving statistical production needs.

Chapter 7

Conclusion

This thesis addressed the parameterization of the Generalized Fellegi-Holt (GFH) framework by developing a data-driven methodology for generating reliability weights. The objective was to replace predefined, static weights with dynamic, record-specific weights derived from machine learning models that identify complex error patterns. The central hypothesis was that this data-driven approach would better align automated error localization with corrections made by human experts.

To test this hypothesis, a methodology was implemented using manually corrected data from the CBS "Structural Business Statistics Survey". Pairwise co-editing patterns were identified with Chi-square tests, while higher-order error patterns were found using the Eclat algorithm and validated with log-linear models. Logistic regression models were then trained to predict these error patterns based on business characteristics. These models produced the record-specific weights for the final hierarchical system (Version 7) used in the MIP solver.

The empirical evaluation confirmed the hypothesis, as all versions with empirical weights (2–7) outperformed the static baseline (Version 1). Version 5, which added dynamic record-specific weights, improved both precision and recall while maintaining efficient processing times (median duration < 0.13 seconds per record). The most comprehensive model, Version 7, integrated dynamic and hierarchical weights, producing the highest F1 scores and true positive counts. This improvement in recall, however, corresponded with a reduction in precision and a doubling of the median processing time compared to non-hierarchical models. This result highlights a trade-off: enabling more generalized corrections increases error coverage but also the risk of over-correction.

The evaluation of the model versions clarifies that the choice of an "optimal" system for deployment depends on multiple factors beyond a single performance metric. The selection of a production-ready model involves balancing performance against operational constraints. While the hierarchical model (Version 7) is the most robust in a technical sense, its significant compu-

tational cost and lower precision present practical challenges. For NSIs like CBS, high precision is often critical to avoid false positives, which can trigger unnecessary manual interventions and reduce the efficiency of the editing process. The strong performance of hybrid models like Version 2 and Version 4 highlights the quality of the initial expert-defined weights.

Therefore, focusing on precision and computational complexity for practical implementation, **Version 4** offers a compelling balance, leveraging expert weights while introducing dynamic pairs at a low computational cost. Alternatively, **Version 6** represents a powerful, fully data-driven approach that includes higher-order patterns without the full overhead of the hierarchical system. Ultimately, the tolerance for false positives may also depend on the quality of the subsequent imputation step; if automatic imputations are highly plausible, a lower-precision, higher-recall model like Version 7 could be viable. The optimal choice requires careful consideration of institutional priorities, computational resources, and operational workflows within specific statistical production environments.

This research demonstrates that integrating machine learning techniques with established statistical frameworks can yield significant improvements in automated data editing while maintaining the interpretability and reliability standards essential for official statistics. The methodology provides a foundation for more intelligent editing systems that adapt to data characteristics and evolve with changing statistical production requirements, ultimately supporting the continued modernization of statistical data processing in national statistical institutes.

Appendix

Table A.1: A simplified, illustrative example of the dataset structure.

Business ID	Year	2D NACE	Size Class[a]	Turnover (€k)	Staff Costs (€k)	Edit Status[b]	Imputed[c]
10114	2022	25	M	12,345	2,150	Auto-Only	False
10287	2022	62	S	890	120	Manual-Only	False
10355	2022	52	L	45,678	11,300	Hybrid	False
10491	2022	41	S	1,250	220	None	True
10523	2022	47	M	2,100	350	None	True

Note: This table presents invented data to illustrate the structure and variable types.
[a] **Size Class**: S=Small, M=Medium, L=Large.
[b] **Edit Status**: The editing process for a responding unit.
[c] **Imputed**: Indicates if the record was imputed due to non-response.

Selected Variable Descriptions

Variable Category / Name	Interpretation
Identifier Variables	
Business ID	A unique numerical identifier for each enterprise.
Group ID	An identifier for the enterprise group a business may belong to.
Classification Variables	
Year	The reporting year for the survey data.
Industry (NACE)	The statistical classification of economic activities.
Size Class	A categorical variable indicating if the business is Small, Medium, or Large.
Financial Variables	
Turnover	Net turnover of the business in thousands of Euros.
Staff Costs	Total personnel costs of the business.
Metadata and Statistical Variables	
Edit Status	A flag indicating how a record was processed (e.g., Auto, Manual).
Imputed	A flag indicating if the record was imputed due to non-response.
Inclusion Weight	The statistical weight of the enterprise in the sample.
Questionnaire ID	An identifier for the specific version of the survey form used.

The tables in Table A.1, A.2, and A.3 present the results of logistic regression models fitted to predict co-editing patterns, where those specific variables are corrected simultaneously during manual editing. The models predict the probability of co-editing events based on business characteristics, with odds ratios indicating the relative likelihood of pattern occurrence, p-values showing statistical significance, and adjusted p-values controlling for multiple testing across all models.

Predictor	Odds Ratio	p-value	Adj. p
PS.INKWRDE100000 & PS.PERSONS110000			
Legal Units: Multi (vs Single)	1.3168770	0.0000001	0.0000004
Size: Large (vs S)	1.2200930	0.0137105	0.0342762

Figure A.1: Snapshot of Model Results for Co-Editing Predictors for Pairs

Predictor	Odds Ratio	p-value	Adj. p
PS.LOONSOM110000 & PS.PERSLST100000 & PS.LOONSOM121200			
Legal Structure: Natural Person	0.4878610	0.0000001	0.0000006
Legal Units: Multi (vs Single)	1.3466738	0.0000008	0.0000020
Size: Large (vs S)	0.7594173	0.0030361	0.0049954
New Business: Yes (vs No)	1.4798534	0.0039963	0.0049954

Figure A.2: Snapshot of Model Results for Co-Editing Predictors for Triplets

Predictor	Odds Ratio	p-value	Adj. p
PS.PERSONS110000 & PS.SUBTOWP200000 & PS.PERSONS100000 & PS.RESULTS120000			
Legal Units: Multi (vs Single)	1.3358574	0.0000023	0.0000113
Size: Medium (vs S)	1.4624468	0.0000443	0.0001107
Size: Large (vs S)	1.2558928	0.0233236	0.0388727

Figure A.3: Snapshot of Model Results for Co-Editing Predictors for Quadruplets

Industry Pattern Analysis

Details about variable meanings from the "Structural Business Statistics Survey" can be found at Statistics Netherlands (CBS) (2024).

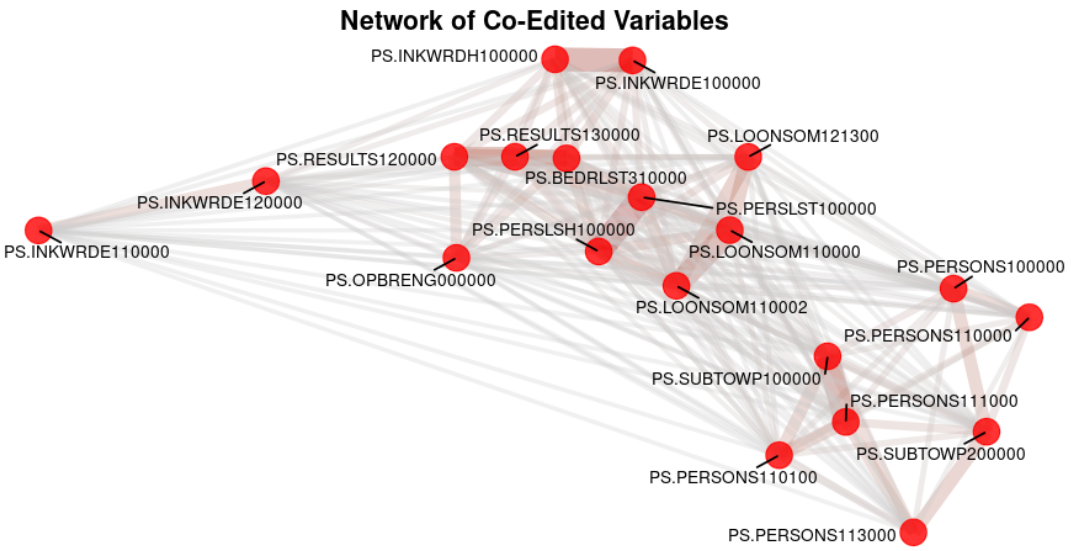


Figure A.4: Network of Co-Edited Variables (with original variable names)

The significant associations graph in Figure A.5 displays the results of Chi-square tests of independence performed on pairs of the top 20 most frequently edited variables, where each tile represents a variable pair and the color intensity corresponds to the negative log-transformed adjusted p-value (using Benjamini-Hochberg correction). The darker colors indicate stronger statistical evidence against the null hypothesis of independence, revealing which variables are most likely to be co-edited together rather than edited independently, with only statistically significant associations ($p_{adj} < 0.05$) displayed in the visualization.

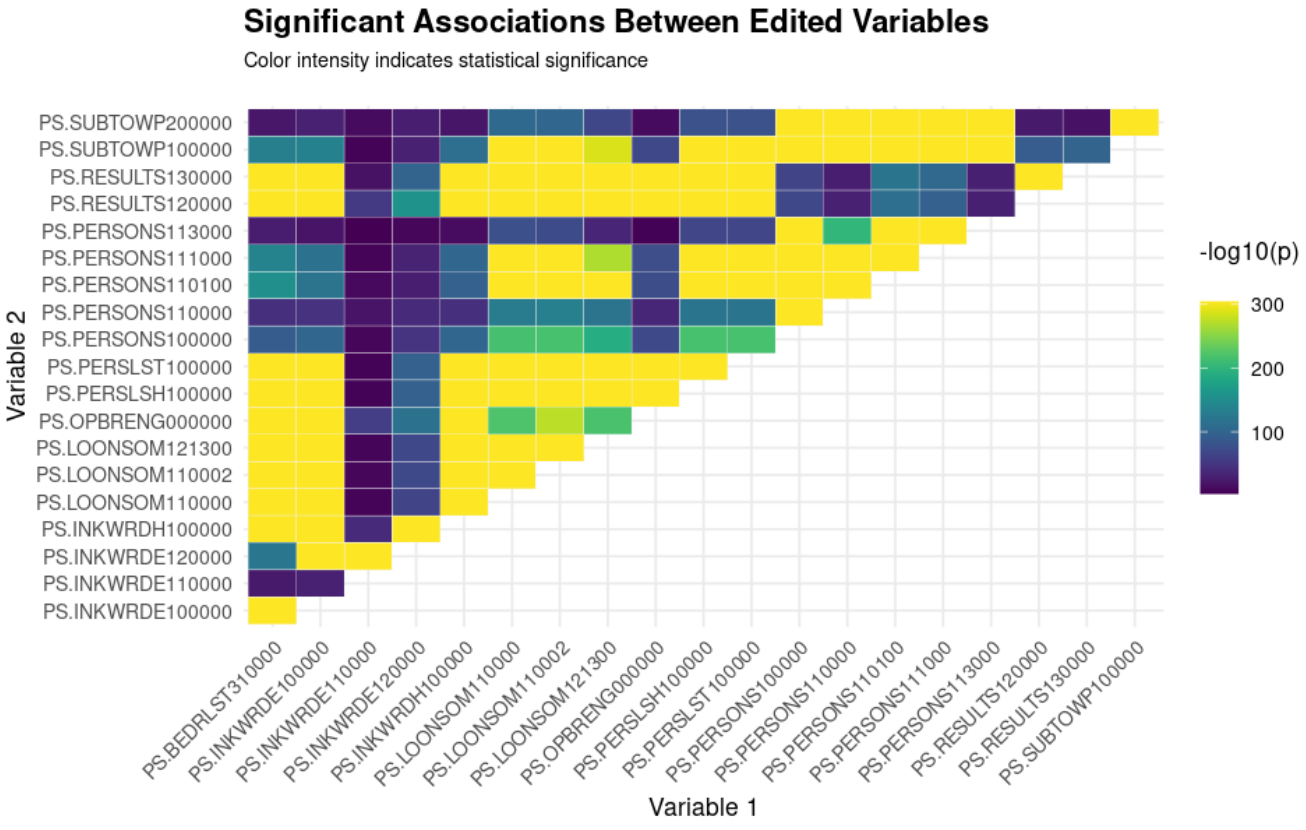


Figure A.5: Significant Association Between Edited Variables

The box-plot in Figure A.6 displays the distribution of edit rates across business sizes (Small, Medium, Large) using a logarithmic scale to handle the wide range of values and compress outliers. Each box represents the distribution of edit rates for all variables within that size category.

Figure A.6: Edit Rate Distribution by Business Size

The heatmap in Figure A.7 displays edit rates for the top 15 most frequently edited variables across different NACE (industry classification) codes, where color intensity represents the frequency of manual corrections within each industry-variable combination.

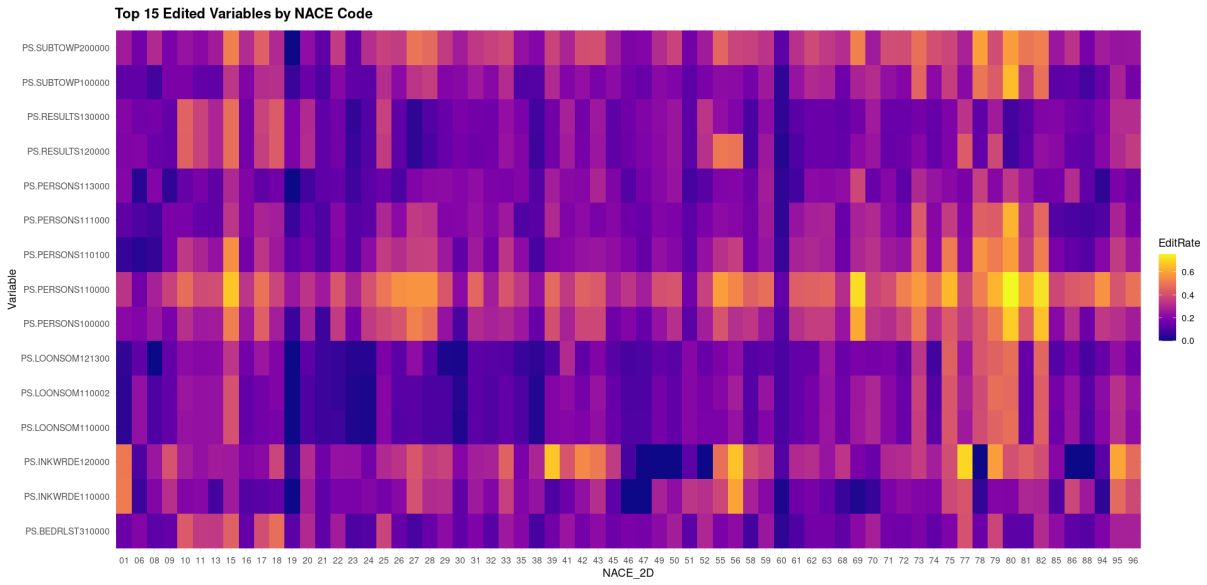


Figure A.7: Top 15 Edited Variables by Nace Code

The bubble chart in Figure A.8 displays aggregated edit rates by NACE industry code and variable groups (identified

by the initial letters of variable names using regex pattern matching), where bubble size represents the maximum edit rate within each industry-variable group combination and color distinguishes different variable categories.

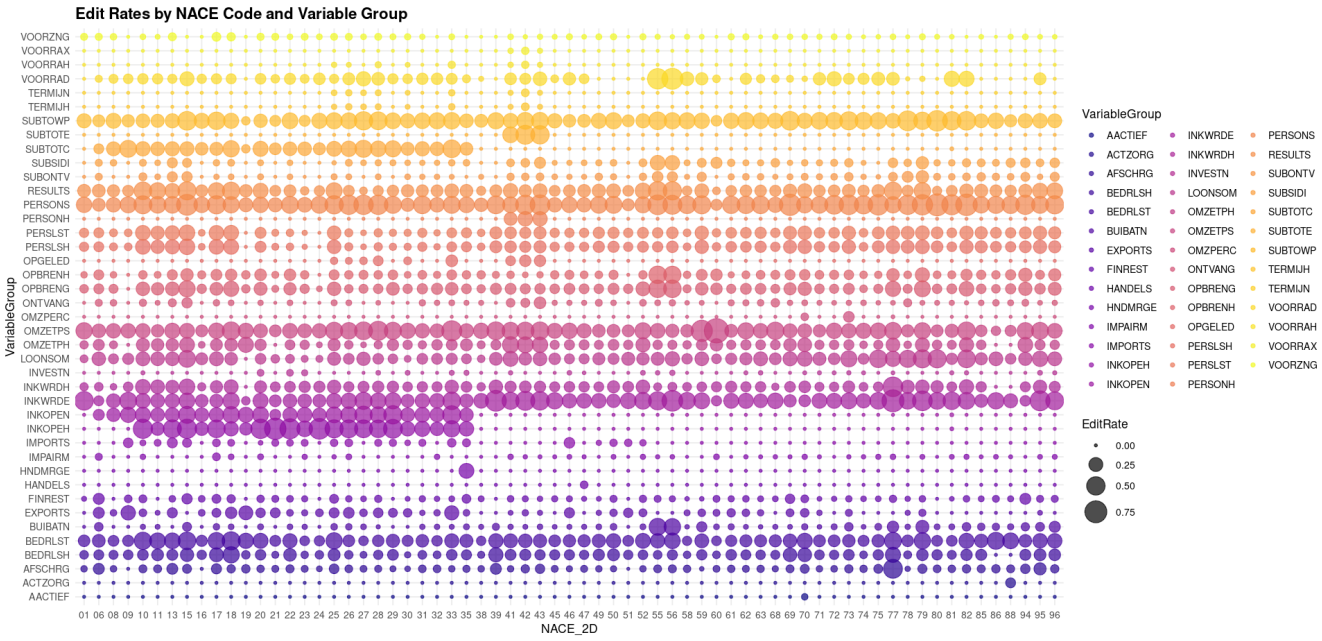


Figure A.8: Edit Rates by Nace Code and Variable Group

The horizontal bar chart in Figure A.9 displays the top edit rate differences between natural persons and legal entities across survey variables, where bar direction indicates which business type is more likely to require manual corrections and bar length represents the magnitude of the difference. Variables extending to the right show higher edit rates for natural persons, while those extending left indicate legal entities require more corrections.

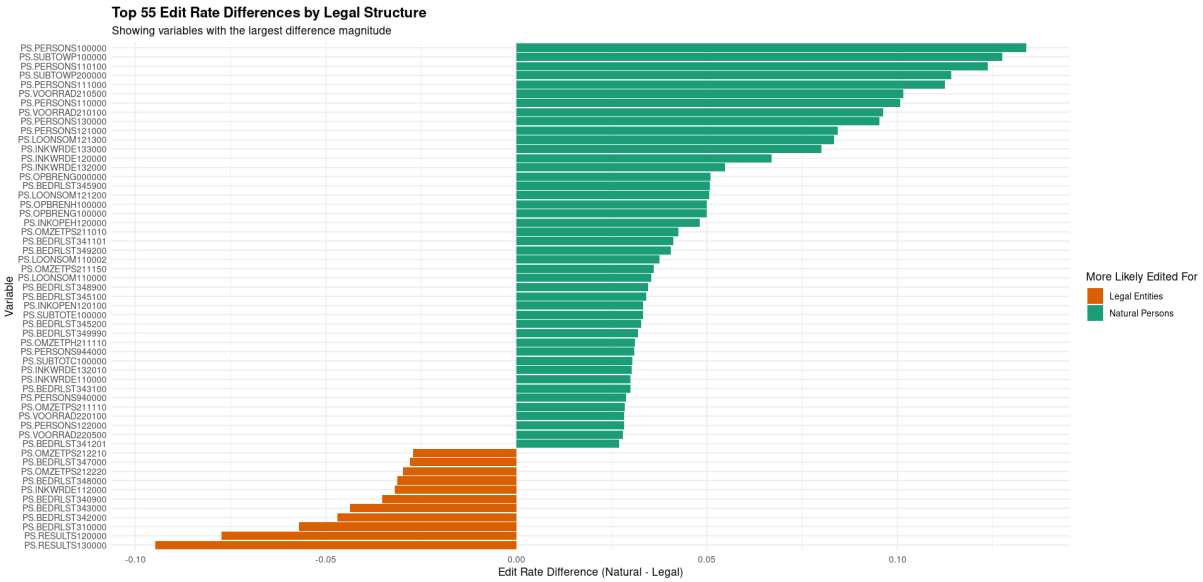


Figure A.9: Top 55 Edit Rate Differences by Legal Structure

Bibliography

Beck, M., Dumpert, F., and Feuerhake, J. (2018). Machine learning in official statistics. *Statistisches Bundesamt, Wirtschaft und Statistik*, 6:33–44.

Daalmans, J. and Scholtus, S. (2018). A MIP approach for a generalised data editing problem. Discussion paper, Statistics Netherlands, The Hague.

de Jonge, E. and van der Loo, M. (2014). *editrules: Parsing, Applying, and Manipulating Data Cleaning Rules*. CRAN. R package version 2.9.5.

de Waal, T., Pannekoek, J., and Scholtus, S. (2011). *Handbook of Statistical Data Editing and Imputation*. Wiley Handbooks in Survey Methodology. John Wiley & Sons, New York.

Fellegi, I. P. and Holt, D. (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71(353):17–35.

Garfinkel, R. S., Kunnathur, A. S., and Liepins, G. E. (1986). Optimal imputation of erroneous data: Categorical data, general edits. *Operations Research*, 34(5):744–751.

Liepins, G. E. (1980). A rigorous, systematic approach to automatic data editing and its statistical basis. Report ORNL/TM-7126, Oak Ridge National Laboratory.

Pannekoek, J., Scholtus, S., and van der Loo, M. (2013). Automated and manual data editing: A view on process design and methodology. *Journal of Official Statistics*, 29(4):511–537.

Scholtus, S. (2016). A generalized fellegi-holt paradigm for automatic error localization. *Survey Methodology*, 42(1):1–18.

Statistics Netherlands (CBS) (2024). Ps onderwijs – productiestatistiek onderwijs (microdata). Microdata description page, retrieved 25 June 2025.

van der Loo, M. P. J. and de Jonge, E. (2018). *Data Validation and Localizing Errors in Data Records*. John Wiley & Sons.

Wallgren, A. and Wallgren, B. (2014). *Register-based statistics: Statistical methods for administrative data*. John Wiley & Sons.

Williams, H. P. (2013). *Model Building in Mathematical Programming*. John Wiley & Sons, Chichester, West Sussex, UK, 5th edition.

Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1999). New algorithms for fast discovery of association rules. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, pages 283–286.