One-Stop-Shop Artificial Intelligence and Machine Learning for Official Statistics

Project 101146355 – AIML4OS

Workpackage 9

USE CASE: IMPUTATION FOCUS – STATISTICALLY VALIAND EFFICIENT EDITING AND IMPUTATION IN OFFICIAL STATISTICS AI/ML – WITH A SPECIAL FOCUS ON IMPUTATION

Deliverable	9.1 – Methodological aspects from use cases in Machine Learning techniques for early imputation in the production of official statistics.
Month due	M 2024
Type	REPORT
Prepared by	Sandra Barragán Andrés (Statistics Spain, INE)

Workpackage Leader:

NAME Sandra Barragán Andrés
EMAIL ADDRESS sandra.barragan.andres@ine.es

Participants in the projects of this deliverable

Spain

Instituto Nacional de Estadística (INE)
Sandra Barragán Andrés
Elena Rosa Pérez
José Manuel Martín del Moral
Beatriz Acereda Serrano

Italy

Istituto Nazionale di Statistica (Istat)
Romina Filippini
Fabrizio De Fausti
Simona Toti

Germany

Statistisches Bundesamt (Destatis)
Kerstin Lange
Steffen Moritz

Poland

Główny Urząd Statystyczny (GUS)
Jarosław Napora
Sebastian Wójcik

Contents

Executive summary

This document addresses the methodological outcomes in relation to the final aim of carrying out early estimation by completing the whole set of microdata through imputation. This imputation could be done with different techniques, here we evaluate the use of machine learning techniques. Different projects carried out by the different participants on this WP9 in relation to the research line about early imputation are shown as particular cases.

This document is divided in six chapters. The Chapter 1 includes the introduction with a brief background on the used techniques can be read. The Chapter 2 has the descriptions of of the four projects related to early imputation in WP9 (Italy, Germany, Spain and Poland). The Chapter 3 contains all the information and suggestions about the preprocessing of the data including cleaning and feature engineering. The Chapter 4 consists of the issues related to model selection: algorithms and hyperparameters to select the best model, train and test steps and model evaluation at micro and macro levels. In the Chapter 5 the results of each project are included. Finally in the Chapter 6 and last chapter some conclusions can be found regarding methodological outputs, the production focused strategy and some final remarks. At the end of the document, Appendix A contains the information about the specific regressors built in each project and in Appendix B some additional tables with results.

1

General Introduction

The modernization of the production of official statistics is rooted on three basic pillars, namely: (i) industrial standardization, (ii) new data sources, and (iii) new statistical methods. The industrial standardization comprises the adoption of international production standard models such as the Generic Statistical Business Process Model, GSBPM for short, [UNECE, 2025], the Generic Statistical Information Model, GSIM for short, [UNECE, 2019b], the Generic Statistical Activity Model for Statistical Offices, GAMSO for short, see UNECE [2019a] and UNECE [2021a]. The UNECE Generic Statistical Data Editing Model (GSDEM) [UNECE, 2019c] plays a pivotal role in official statistics by providing a standardized conceptual framework for data editing and imputation. Its importance lies in harmonizing terminology and processes across National Statistical Institutes, facilitating the transition from ad-hoc manual corrections to systematic, reproducible, and automated imputation workflows. By aligning with the GSBPM, the GSDEM ensures that imputation techniques are integrated into a transparent quality management cycle, ultimately enhancing the reliability and comparability of statistical outputs.

The core aspect of this industrial standardization and these models is the adoption of a modular approach to statistical production and an enterprise architecture (see e.g. Eurostat [2019a] for an European context). The incorporation of new data sources comprises the use of traditional survey data together with administrative registers and new digital data of any nature. This incorporation entails a fairly large amount of issues to be solved [see e.g. Kitchin, 2015, Hand, 2018, Salgado and Oancea, 2020, and multiple references therein], which is intimately linked to the need of using new statistical methods and a refurbished production framework.

One of the potentially ensuing advantages of this modernization process is the generalized improvement of quality, understood in its multidimensional conception [see e.g. Wand and Wang, 1996, ESS, 2014]. According to most appraisals, the huge availability of data is expected to remarkably improve the timeliness quality dimension, bringing dissemination of results closer to the point in time when phenomena take place. This timeliness improvement is usually associated to new data sources, especially, digital data sources. The initial success of the example of the detection of influenza epidemics using search engine query data, i.e. of the so-called Google Flu Trends [Ginsberg et al., 2009] spread the feeling that digital data could bring immediacy to the production of statistics. However, it was soon demonstrated that this type of analysis is prone to significant inaccuracies [D. Lazer et al., 2014]. The faintness of many of these correlations between digital data and the target variables in Official Statistics arises mostly due to the lack of statistical structural medatata (concepts and definitions) in all these new data sources.

The core objective of this report is to explore methodologies for producing early estimates, or carrying out nowcasting, of key official statistics. Our specific focus is on the microdata

approach, which aligns with the preferred methodology of leading statistical institutes [Eurostat, 2017]. This approach conceptualizes nowcasting not as the direct prediction of an aggregate figure, but as the *early reconstruction of the underlying microdataset*. In this framework, the nowcasting problem is fundamentally recast as a large-scale imputation challenge: for the most recent reference period, a significant portion of the target microdata is missing (due to reporting lags), and the task is to impute these missing values for individual statistical units using all available early signals and auxiliary information. Once a complete (albeit partially imputed) microdataset is obtained, the nowcasted aggregate indicator can be derived using standard compilation formulas, ensuring methodological consistency and also coherence between the different breakdowns (regional, economic, etc.) with the final published statistic. It is precisely within this challenging context of high-dimensional, mixed-type microdata with complex missing patterns that Machine Learning (ML) imputation techniques demonstrate their particular promise.

The projects presented in this report will investigate and evaluate specific ML-based imputation strategies for nowcasting, examining their operational feasibility, accuracy, and alignment with the quality frameworks of official statistics.

1.1. Early estimation in Official Statistics

The provision of timely and accurate information is crucial to have knowledge about the present state of the economy and of society, in general UNECE Machine Learning Project Team [2020] and Broe et al. [2021]. This is fundamental for a more efficient policy-making and decision-taking at all levels. In this sense, the subject *nowcasting/early estimates* has received much attention in the past years, moving beyond traditional dynamic factor models to incorporate machine learning and high-frequency alternative data [see e.g. Giannone et al., 2013, Bok et al., 2017, OECD, 2023, Kant et al., 2024, Dorville et al., 2025, Eurostat, 2025, and multiple references therein].

Terms to refer to this type of estimates abound in the literature: early estimates, rapid estimates, flash estimates, nowcasting, forecasting, leading indicators, coincident indicators, advance estimates,... with subtle differences between them (sometimes none).

A specific and critical application where imputation techniques, and increasingly ML, play a pivotal role is in the production of *nowcasts* or early estimates. Early estimation, also known as nowcasting, refers to the prediction of the present, the very recent past, or the near future of a key economic or social indicator, bridging the publication lag inherent in traditional statistical processes [Eurostat, 2017]. The demand for more timely indicators from policymakers and the public has made nowcasting a priority for many NSIs [Lotrič Dolinar et al., 2025].

In the production of official statistics, timeliness is not merely a logistical goal but a core dimension of quality. The relationship between the time of release and the utility of the data can be conceptualized through a trade-off model between two competing forces [Eurostat, 2021] (see Figure 1.1):

- **Relevance and Utility (Descending Curve):** Statistical information is a "perishable good." Immediately after the reference period, its value for decision-making is at its peak. As time passes ($t \to \infty$), the relevance of the data decays because the socio-economic reality it describes has already changed.

- **Accuracy and Precision (Ascending Curve):** At early stages, data often relies on preliminary estimates, incomplete surveys, or proxy indicators, leading to higher variance. Accuracy increases as more reliable auxiliary information is integrated and validation processes are completed.

The **Optimal Release Point** is defined as the temporal equilibrium where the marginal gain in accuracy no longer compensates for the marginal loss in relevance. Publishing after this point results in highly precise data that is historically significant but operationally obsolete.

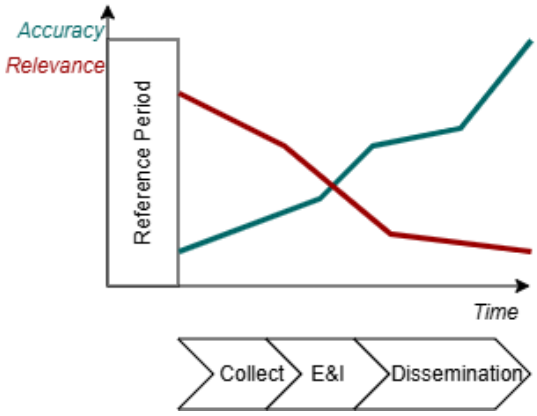

Figure 1.1 Trade-off accuracy vs. relevance in the production process

The importance of timeliness is formally recognized by international frameworks such as the IMF's Data Quality Assessment Framework (DQAF) under the dimension of *Serviceability* International Monetary Fund [2012]. The technical necessity of prioritizing timeliness is based on three pillars:

1. **Evidence-Based Policymaking:** For statistics to serve as a diagnostic tool (e.g., CPI for monetary policy or GDP for fiscal adjustment), they must be available within a window that allows for intervention. Delayed statistics turn proactive policy into reactive history.

2. **The Cost of Information Asymmetry:** If the National Statistical Office (NSO) fails to provide timely data, the market fills the void with private estimates that may lack the methodological rigor and transparency of official statistics, potentially leading to misinformation [Holt, 2008].

3. **The Quality Frontier:** High-quality statistics must find the "efficiency frontier." An obsession with zero-error margins can be counterproductive; if a census or a major survey takes several years to be published, the structural changes in the population may render the results non-representative of the current state.

Two primary methodological approaches dominate nowcasting in official statistics, differing in their level of data aggregation:

1. **Microdata Approach (Preferred):** This method involves predicting the missing values for individual units (e.g., companies, households) in the most recent period for which the target survey or register is incomplete. Once these unit-level values are imputed, the aggregate indicator (e.g., GDP growth, unemployment rate) is calculated from the completed microdataset using the standard aggregation formula. This approach is generally preferred by statistical institutes for several reasons [UNECE, 2021b]:

- **Methodological Consistency:** The nowcast is derived using the same concepts, definitions, and compilation rules as the final official statistic, ensuring coherence.
- **Transparency and Auditability:** The imputation model can be validated at the unit level, and the contribution of each unit to the final nowcast is traceable.
- **Rich Information Use:** It can incorporate a wide array of auxiliary variables available at the micro level (e.g., other survey variables, high-frequency administrative data) to inform the prediction for each unit.
- **Flexibility:** It allows for the calculation of not only the aggregate figure but also its distribution and breakdowns by relevant subpopulations from the same completed dataset.

The challenges of this approach include the need for high-quality, timely auxiliary microdata and the computational complexity of running imputation models on large-scale datasets.

2. **Aggregate Indicator Approach:** This alternative models the target aggregate indicator directly as a time series. It uses past values of the indicator itself, combined with other related aggregate time series (e.g., financial indices, sentiment indicators, industrial production) that are available with a shorter lag, to predict the most recent value of the target indicator. Techniques range from traditional time series models (e.g., ARIMA, VAR) to more advanced ML models like Random Forests or LSTM neural networks [Bok et al., 2018]. This approach is often simpler and faster to implement. However, this approach deviates from usual compilation methodologies, and alignment with official definitions relies just on its predictive value. It does not leverage the full richness of microdata and can be harder to integrate transparently into the official production process.

The evolution of nowcasting is increasingly leaning towards hybrid or ensemble models and the integration of ML techniques, particularly within the microdata framework. ML algorithms are well-suited to handle the non-linear relationships and high-dimensional auxiliary data (e.g., scanner data, web scraped prices, satellite imagery) that can improve the accuracy of unit-level imputations for early estimates. As with imputation, the challenge lies in implementing these methods in a manner that is verifiable, documented, and consistent with the quality framework of official statistics.

1.2. Imputation in Official Statistics

The compilation of high-quality, complete, and coherent datasets constitutes a fundamental challenge in the production of official statistics. Item non-response, whether collected from survey participants, administrative registers, or integrated data sources, introduces missing values that can compromise the validity of statistical inferences, bias key indicators, and reduce the effective sample size [Eurostat, 2018]. Imputation, defined as the process of replacing missing values with plausible substitutes, is therefore an indispensable step in the statistical data processing chain. Its primary objectives are to reduce non-response bias, to restore the completeness of datasets for subsequent analyses that require complete records (e.g., multivariate modeling), and to ensure consistency across aggregated figures and published tables [United Nations Economic Commission for Europe (UNECE), 2006]. Standard practices for ensuring data consistency, ranging from deterministic editing to stochastic imputation models, are extensively detailed by de Waal et al. [2011].

In the context of official statistics, imputation is not merely a technical procedure but is deeply intertwined with principles of transparency, reproducibility, and the preservation of data integrity. Traditional imputation methods employed by National Statistical Institutes (NSIs) have largely been deterministic (e.g., deductive, mean/median imputation, ratio imputation) or model-based, predominantly using regression and donor-based methods such as hot-deck and nearest neighbour imputation [Särndal et al., 1992]. These methods are often governed by well-documented frameworks, such as the recommendations on Luzi et al. [2007], which emphasize the importance of preserving the distribution of the observed data and the relationships between variables. The validity of these classical approaches, however, often rests on assumptions (e.g., Missing Completely at Random – MCAR, or Missing at Random – MAR) and model specifications that may become untenable in the face of complex, high-dimensional data structures increasingly common in modern statistical production, see Eurostat [2014].

1.3. Machine Learning in Official Statistics

Modern machine learning algorithms are built upon foundational mathematical concepts that bridge data analysis and optimization [James et al., 2013]. Here, our focus lies entirely within the domain of supervised learning. Since it is illustrated in the Figure 1.2 from Chollet and Allaire [2018], the paradigm from classical programing has changed with the machine learning models. In classical programming, rules and data are combined to produce answers, whereas in machine learning, answers and data are used to learn the rules.

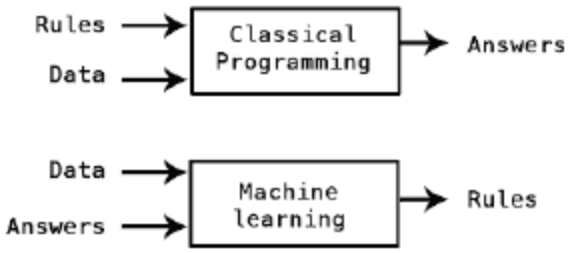

Figure 1.2 Classical programming vs. machine learning [Chollet and Allaire, 2018].

The formula of a general model of statistical learning is:

$$Y = f(X) + \epsilon,$$

where Y is the target, X are the regressors, f is the model and ϵ the error. In relation to the target, it can be numerical or categorical, it depends on the goals of each project. Then, the problem can be about regression (numerical target) or classification (categorical target).

The workflow in Figure 1.3 represents a standard end-to-end Machine Learning pipeline designed for modular statistical production, emphasizing the transition from raw data ingestion to model deployment. Initially, feature engineering (details in chapter 3) is executed to transform raw administrative or survey data into a structured format enriched with regressors. This processed dataset is then partitioned during the Split phase into training, validation, and testing sets to prevent data leakage. The core of the model selection process involves cross-validation using the training and validation subsets to iteratively evaluate performance and identify the optimal model architecture and hyperparameter configuration (more details in chapter 4). Once the best model is selected, it undergoes a final training phase—often using the combined training and validation sets, to produce the final trained model. Finally, the model's

generalization capabilities are rigorously assessed against the independent test data, ensuring that the resulting statistical estimates are both accurate and reliable for official dissemination. This process ensures the quality of the final results. The deployment in production of these kinds of procedures should be done with specific tools and a well-designed implementation, see Burkov [2020].

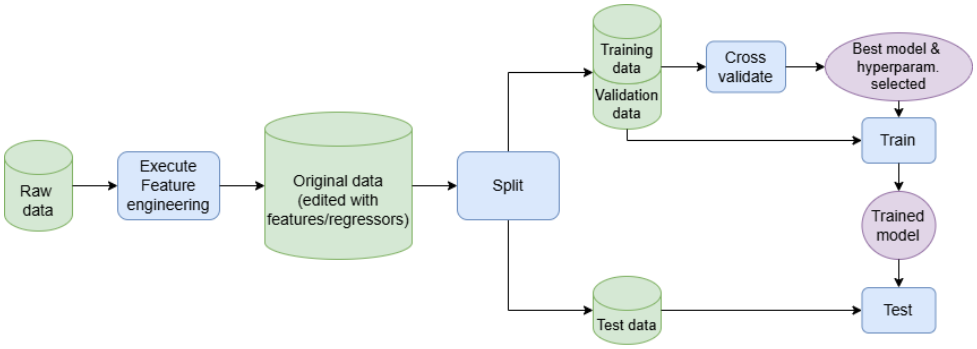

Figure 1.3 Machine learning generic process

The advent of the data revolution has prompted official statistics to explore innovative methods to enhance the efficiency, granularity, and timeliness of its outputs. Machine Learning (ML), a broad set of algorithms capable of identifying complex patterns and relationships from data, is gaining traction within NSIs [UNECE, 2021b]. Applications range from nowcasting to the classification of textual responses and the detection of outliers in data validation processes. The appeal of ML lies in its ability to handle large volumes of data with numerous, potentially non-linear, interactions without the need for strong *a priori* model specifications.

The adoption of ML in official statistics is cautious. Key documents, such as the UNECE's *Machine Learning Task Team* reports, underscore the necessity of aligning ML applications with the fundamental principles of official statistics, including objectivity, reliability, and the explicability of methods [UNECE, 2021b]. The "black-box" nature of some advanced algorithms poses challenges for transparency and quality measurement, which are paramount in an official statistical context. Consequently, the focus is often on *supervised* and *unsupervised* learning techniques that can be integrated into existing production systems while allowing for rigorous validation and explanation of their outputs. As documented in the HLG-MOS Machine Learning Project's final report [UNECE Machine Learning Project Team, 2020] for Work Package 1 [Dumpert, 2020], several National Statistical Offices are transitioning from rule-based editing to more automated, ML-driven approaches. The results from various pilot studies compiled suggest that supervised learning models, such as Random Forests, can effectively identify outliers and impute missing values with high accuracy.

Dumpert [2025] provides a critical and comprehensive examination of the integration of machine learning methodologies into the established paradigm of official statistics. It moves beyond purely technical exposition to situate this integration within the broader evolution of the statistical industry, addressing its potential to enhance data production efficiency, timeliness, and quality in response to modern societal demands. The work rigorously explores both foundational methodological challenges—such as uncertainty quantification and error frameworks—and practical advances through international case studies. Crucially, it balances this promise with a necessary discourse on the accompanying ethical, legal, and governance imperatives required to preserve the core principles of transparency, fairness, and public trust that underpin official statistical systems.

The application of Machine Learning to the specific problem of imputation represents a natural and promising convergence of the two aforementioned domains. ML-based imputation techniques aim to leverage the pattern-recognition strength of algorithms to generate more accurate and realistic imputations, particularly when the missing data mechanism is complex or the data contain intricate interactions.

These methods can potentially outperform classical techniques when the relationships between the target variable and the auxiliary information are non-linear or involve high-order interactions. As reviewed by several authors [see e.g. Prusty et al., 2020, Emmanuel et al., 2021, Adithya and Ancy, 2022, and multiple references therein], machine learning and AI-based imputation techniques, such as K-Nearest Neighbors (KNN), Random Forests, and Artificial Neural Networks, offer significant advantages over traditional statistical methods by capturing complex non-linear relationships within the data. However, their implementation in official statistics requires careful consideration. Critical issues include the computational cost for large-scale data, the risk of overfitting, the handling of categorical and mixed data, and, most importantly, the development of appropriate variance estimation and quality measures for the imputed values.

Research documented in the *Journal of Official Statistics* and UNECE working papers is increasingly focused on these operational and methodological challenges, seeking to bridge the gap between ML's predictive power and the rigorous quality standards mandated for official statistical production [High-Level Group for the Modernisation of Official Statistics (HLG-MOS), 2024]. As highlighted by Yung et al. [2022] in their comprehensive quality framework, statistical algorithms must be assessed not only on accuracy but also on dimensions such as explainability, reproducibility, and timeliness.

Machine Learning algorithms significantly enhance data quality and decision-making, in particular in the application to imputation, although their effectiveness varies depending on the specific domain and data type. Deep Learning (DL) is widely recognized for its accuracy and efficiency in handling complex relationships in large-scale data. However, it is less frequently employed in official statistics, as survey microdatasets are often not large enough to gain a competitive advantage over tree-based methods. Additionally, DL tends to be less effective when applied to structured tabular data—the standard format in this field—relative to its performance on unstructured data types.

Finding the best algorithm is not an easy task. To guide this process, Figure 1.4 by Alabadla et al. [2022] introduces a structured taxonomy designed to facilitate the selection of the most appropriate imputation method based on the specific features and requirements of the data.

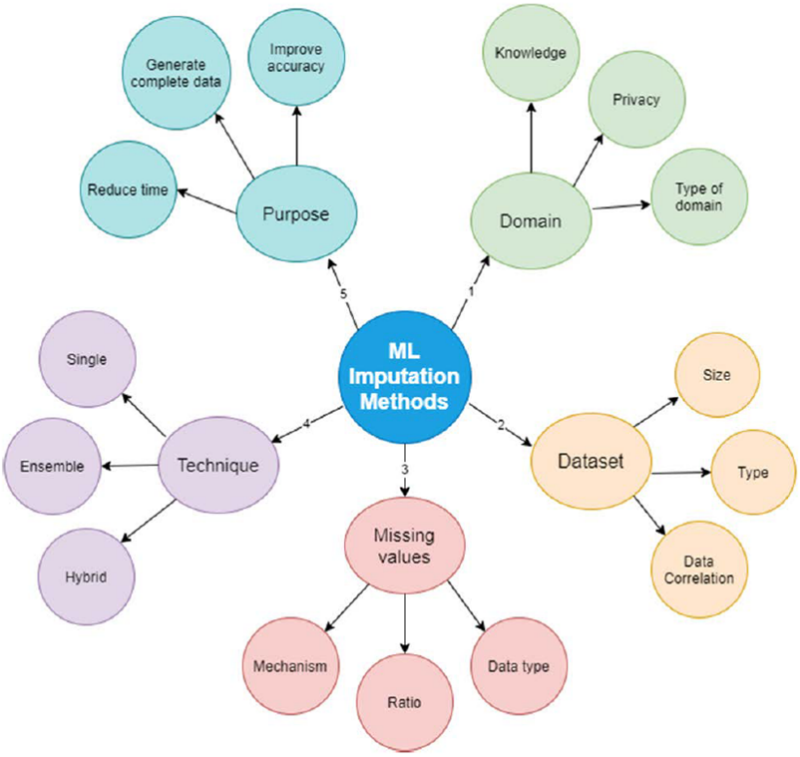

Figure 1.4 The proposed taxonomy for selecting the best ML imputation method by Alabadla et al. [2022].

The remainder of this report is structured as follows. To provide a clear roadmap of the research about early imputation as part of WP9, chapter 2 introduces the four national projects, followed by chapter 3, which details the data cleaning and feature engineering processes. Chapter 4 describes the machine learning process core, from algorithm and hyperparameters selection to model evaluation. The findings are presented in chapter 5, and the final conclusions and production strategies are discussed in chapter 6. Additional data on specific regressors and supplementary results can be found in the Appendices A at the end of the document. This document presents an harmonized methodology of the four projects into a single, cohesive structure. This report details all the methods required to understand our practical implementation, including essential basic concepts. However, the document is intentionally focused on the techniques actually used; therefore, we do not delve into methods that fall outside the scope of these specific projects.

2

Projects Description

In this chapter the four projects related to early imputation are described. The notation regarding the reference period could appear in two equivalent forms: $m + xd$ represents x days after the reference month m or more general $t + x$ with t the reference period. The rest of the notation could change among projects. First a brief summary of the projects:

Italy (IT-SchoolEnrollment): The primary objective of this project is to provide one-year forward projections for school enrollment by leveraging demographic and historical administrative data. The study focuses on predicting the enrollment status of individuals, a **categorical target variable**, within the primary and secondary education levels. The underlying data and the resulting estimates follow an **annual** periodicity.

Germany (DE-ITI): This project aims to accelerate the publication of the Industrial Turnover Index (ITI) by providing early estimates as soon as 15 to 20 days after the reference period. The methodology focuses on imputing the **numerical values** of monthly turnover for units that have not yet reported their data. The survey is conducted on a **monthly** basis.

Spain (ES-ITI): Similar to the German case, the Spanish project focuses on the Industrial Turnover Index, with the goal of producing daily flash estimates and reconstructing the complete microdataset. The target variable is the **numerical** monthly turnover of industrial establishments. The survey follows a **monthly** periodicity, with data being processed in several batches throughout the collection period.

Poland (PL-Accommodation): The objective of this project is to develop flash estimates for tourist accommodation occupancy to address delays in data submission. The model specifically targets the total number of tourists, a **numerical** variable, to mitigate the impact of varying response rates. The survey is carried out with a **monthly** periodicity and exhibits strong seasonal patterns.

2.1. Early imputation in school enrollment (IT) [IT-SchoolEnrollment]

2.1.1. Description

A high amount of information on education (such as school enrollment and educational attainment) is available from administrative sources. However, these sources often present critical aspects related to coverage and timeliness. Regarding the educational level, a procedure has already been developed, based on the integration of administrative and survey data. The procedure is currently used for producing census outputs. In this project the production of estimates on school enrollment, based on the use of administrative sources is studied. In particular, the case study addresses the issue of timeliness by aiming to predict school enrollment for year t based on demographic characteristics and longitudinal data on education available from administrative sources up to year t-1.

2.1.2. Use of Machine Learning

Machine Learning (ML) methods can leverage all available information, including historical data and variables with many categories, capturing complex relationships between variables, which is often challenging to incorporate comprehensively using standard methods. The goal is to evaluate the accuracy of results, mainly in terms of distributions (macro-accuracy), achievable with ML methods.

The experimentation is carried out on past data so that the true value of the response variable (school enrolment) is available from administrative sources. This allows comparisons of the estimated value with the true value in both micro and macro level.

On the same experimental dataset, a standard method, specifically logistic regression, is applied, so that the real value added in the use of ML methods, with respect to standard method, is evaluable in terms of both result accuracy and process efficiency.

2.1.3. Exploratory Data Analysis

Description of the input data:
Variables related to school enrollment that can be considered as input data are:

- demographic characteristics: age, gender, citizenship, region, province and municipality of residence;
- information on school enrollment, available from the 2015/2016 school year up to 2020/2021 (t-2/t-1) school year: school enrollment status, and, if enrolled, year of attendance, type of attendance (full school year, partial school year, school change during the year), municipality of the attended school, and school identification code.

The school identification code consists of 10 digits with a fixed structure, where each digit or subset of digits carries specific meaning. To maximize the utility of this information as input for the models, separate variables were derived by decoding the components of the school identification code. Specifically:

- `digit12`: province where the school is located
- `digit3`: type of school: State or privately subsidized (paritarian - only for primary and lower secondary)

- `digit34`: type of institution (e.g., Lyceum, Technical Institute, etc.) and its educational track (only for upper secondary)
- `digit5`: indicator flag for prison school
- `digit8`: indicator flag for evening school (primarily for adult learners).

Data Analysis:
Since various sources of information are available, a first step of data integration is needed. In particular, the reference population comes from the Italian Base Register of Individuals (BRI), which contains some core variables like place and date of birth, gender and citizenship referred to people resident in Italy in year t. To this reference population, administrative information on education from the Ministry of University and Research (MUR), is added.

School enrollment in year t (corresponding to the school year t/t+1) is the variable of interest. It must be produced by Istat by December of year t+1. Demographic characteristics from administrative sources, for year t, are available, along with longitudinal information on school enrollment up to year t-1 (school year t-1/t). As an example: by December 2024, student estimates for the year 2023 must be produced. The most up-to-date administrative data, available in September 2024, refers to 2022 (school year 2022/2023). Therefore, producing data for 2023 requires a one-year forward projection. All longitudinal information available up to 2022 can, of course, be used as input for the estimation.

In this case study, the reference year t is 2022, which is currently the most recent year with complete information available from administrative sources. This enables a direct comparison between the estimated values and the true values of the variable, allowing for a real accuracy assessment, at both the micro and macro levels. However, the goal of the experimentation is to simulate a real-world scenario, so only administrative information up to 2021 (t-1) is utilized. Given the complexity of the phenomenon under study, arising from the heterogeneity of the reference population and the structure of the available data, we decided to begin our experimentation on school enrollment by focusing on two subsets of the entire population. Specifically, we partitioned the population of interest by school level, and considered the following two subsets of data for the experimentation: 1) Individuals enrolled in primary and lower secondary school; 2) Individuals enrolled in upper secondary school.

Specifically, the aim is to estimate school enrollment in school year 2022/2023, so the criteria considered for selecting the population to study are defined as the union of individuals satisfying at least one of the following conditions:

- enrollment in the relevant school level (primary or lower secondary for the dataset 1; upper secondary for dataset 2) in school year t-2/t-1 (corresponding to the 2020/2021 school year), for the training set;
- enrollment in the relevant school level in school year t-1/t (corresponding to the 2021/2022 school year), for the test set.

2.2. Early imputation in the industrial turnover index in Germany (DE) [DE-ITI]

2.2.1. Description

The $m + 20d$ project on early imputation in the industrial turnover index, conducted by the Federal Statistical Office of Germany, aims to accelerate the availability of short-term economic indicators for the manufacturing sector. To achieve this, microdata-based models are developed, including statistical imputation techniques as well as machine learning approaches, with the objective of producing reliable estimates within 15 to 20 days after the end of the reference month.

A major challenge in this context is the high proportion of missing reports at early stages of data collection: around 50% of reports are not yet available at time $m + 15d$, and approximately 35% remain missing at $m + 20d$. Consequently, model-based imputation is essential to compensate for incomplete data and to ensure the quality of early estimates (see also [Yadegar et al., 2025]).

2.2.2. Use of Machine Learning

The expected value added from Machine Learning in the context of accelerating economic early indicators includes:

1. Faster Data Processing – Reducing the publication time from $m + 45d$ to approximately $m + 15d$ or $m + 20d$.
2. Improved Accuracy – Enhancing economic nowcast with specialized ML models.
3. Higher Efficiency – Automating data integration and statistical compilation, reducing manual effort.
4. Better Decision-Making – Providing policymakers and businesses with timely insights into economic planning.

2.2.3. Exploratory Data Analysis

Description of the input data:
The microdata-based approach relies on data from monthly reports submitted by local units in manufacturing, mining, and quarrying. All producing local units with 50 or more employees are included in this survey, making it effectively a complete survey with a cut-off threshold. In 2024, approximately 23,000 local units participated. The survey collects detailed information on turnover—distinguishing between domestic and foreign turnover as well as main industrial groupings—along with data on products, sales, employment, and hours worked. These data are used to calculate key indicators such as absolute turnover, turnover volume indices, and value indices on a monthly basis. The resulting statistics provide important insights into Germany's economic development and are used by national accounts, the financial sector, ministries, and business associations.

At the lowest level of processing, local units are further subdivided into approximately 32,000 local kind-of-activity units (KAUs). Each local KAU comprises all activities of a local unit that belong to the same group within the classification of economic activities. This level forms the basis for estimation in microdata-based models.

The data used for both regressors and the target variable, namely turnover in manufacturing, extend back to 2014. Up to June 2022, only finalized data for past periods were available, whereas real-time data were not accessible. Since July 2022, however, data have been available in real time, meaning that reports for the current reference month are already accessible during the month. At time $m + 20d$, some local units have not yet submitted their reports, and submitted values may not yet have undergone full data editing. The availability of continuously updated microdata prior to finalization has enabled methodological improvements and supported the development of effective imputation methods within the project (see [Yadegar et al., 2025]). Data are extracted at time $m + 15d$ (until June 2024) and at time $m + 20d$ (on a continuous basis). At these points in time, the data can be divided into three categories: reported values that have already undergone data editing, reported values that have not yet undergone data editing, missing values. These categories must be treated differently in the preprocessing and estimation steps.

Data Analysis:
Figure 2.1 illustrates the share of data available in edited form at time $m + 15d$ and time $m + 20d$, which in turn determines the proportion of data that needs to be imputed. A lower share of edited reports implies a higher need for imputation at the level of local KAUs. Although the group of reporting units is updated only once per year, the availability of edited data for individual local KAUs may vary from month to month. Units that are available at time $m + 15d$ or $m + 20d$ in one month may be missing at the same reference points in another month, and vice versa.

Figure 2.1 Monthly reports cover Sections B and C of the German Classification of Economic Activities, which is based in a legally binding manner on NACE Rev. 2, the European classification of economic activities.

Data availability poses a particular challenge, as the values reported early in the reference month may be systematically biased. Larger and more heterogeneous enterprises tend to report later, meaning that the data available at early stages are not representative of the full population. As a result, these early reports alone are not sufficient for reliably estimating missing values without incorporating additional information from previous reference periods. In this context, a high reporting rate and comprehensive coverage across all German federal states are crucial for producing accurate estimates.

Economic sectors differ substantially with respect to turnover, business structure, volatility, and their share within manufacturing (see Figure 2.2). To account for this heterogeneity, the imputation is carried out separately for each economic activity at the two-digit level, allowing sector-specific patterns to be adequately reflected. The differences in economic importance and size across sectors are evident in the distribution of total turnover as well as in the number of local units within each activity (see also Yadegar et al., 2025).

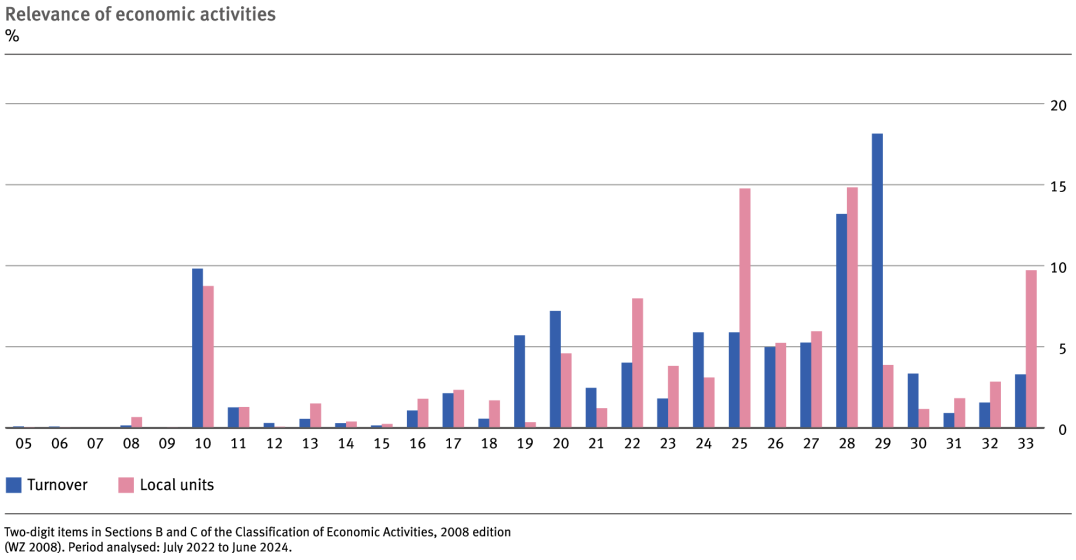

Figure 2.2 Relevance of economic activities

2.3. Early imputation in the industrial turnover index in Spain (ES) [ES-ITI]

2.3.1. Description

The Industrial Turnover Indices (ITI) survey is a European short-term business statistics produced by the European Statistical System (ESS) in compliance with the Regulation (EU) no. 2019/2152 and Comision Implementing Regulation 2020/1197, [Eurostat, 2019b, 2020]. The ITIs have the objective of measuring the evolution of the activity of establishments included in the industrial sector through their turnover. Thus, the core target variable is the *industrial turnover*, i.e. "the value of the invoicing of the establishment, in the reference month, for the sales of industrial goods and provision of industrial services, considering both those carried out by the establishment itself, and those performed through subcontracting with third parties. It therefore includes the income from the sales of finished products, of semi-finished products, of subproducts, of waste and recovered materials, of packages and packaging and of merchandise (goods acquired for resale in the same state as that in which they were acquired), as well as the income from the provision of services related to the normal activity of the establishment" [see INE, 2024, for details].

The ITI survey comprises all industrial establishments whose main economic activity is included in sections B "Extractive industries" (except divisions 06 and 09 in the case of Spain) and C "Manufacturing industry" of the Spanish National Classification of Economic Activities (CNAE-2009), adapted from the international NACE Rev. 2. The survey has monthly periodicity and provides data at national and NUTS2 geographical levels (excluding Ceuta and Melilla). The population frame is built from the Industrial Products Survey[1] and the Structural Business Statistics (Industrial Sector) [2]. Sampling units are selected according to a cut-off sampling design with a sample size of around 12000 units. The sample is revised yearly. The indices follow a fixed-base Laspeyres formula in two steps:

- Computation of the elementary indices. The sample of units is partitioned into strata determined by their NUTS2 geographical variable and some groupings of CNAE-2009 economy activity codes (divisions and aggregation of groups) [INE, 2024]. The elementary index for stratum U_D at reference month m of year y is then computed recursively as

$$I_{U_D}^{my} = \frac{\sum_{k\in S_D^{my}\cap S_D^{m-1y}} z_k^{my}}{\sum_{k\in S_D^{my}\cap S_D^{m-1y}} z_k^{m-1y}} \cdot \ldots \cdot \frac{\sum_{k\in S_D^{2y}\cap S_D^{1y}} z_k^{2y}}{\sum_{k\in S_D^{2y}\cap S_D^{1y}} z_k^{1y}} \cdot \frac{\sum_{k\in S_D^{1y}} z_k^{1y}}{\frac{1}{12}\sum_{i=1}^{12}\sum_{k\in S_D^{iy}\cap S_D^{i-1y}} z_k^{iy}} \cdot 100, \quad (2.1)$$

  where S_D^{my} denotes the sample for the stratum U_D at reference month m of year y (by convention $S_D^{-1y} = U_D$) and z_k^{my} stands for the target variable value of establishment k at reference month m of year y (the total turnover of the industrial establishment) and i stands for the months of the base year.

- Computation of composite indices. After computing the weight w_D^y of each stratum U_D for the base period y using data from the Structural Business Statistics (Industrial Sector), we can compute the composite index for a functional aggregate $U_A = \bigcup_{D\in A} U_D$ just as a weighted arithmetic mean of elementary indices:

$$I_{U_A}^{my} = \sum_{D\in A} w_D^y \times I_{U_D}^{my} \quad (2.2)$$

[1] https://www.ine.es/dyngs/IOE/en/operacion.htm?id=1259931057259.
[2] https://www.ine.es/dyngs/IOE/en/operacion.htm?id=1259931057090.

The main steps in the production pipeline in relation with our pilot study [see INE, 2024, for more details] are:

- Data collection for reference month m starts at $m+1d$ (the immediate day after the reference period ends and the data collected is available on a daily basis).
- Data editing during collection takes place all along the collection period by Statistics Spain's provincial delegations.
- Although the data is available daily, it is processed at Statistics Spain headquarters by the survey managers in three data batches constituted at $m + 20d$, $m + 29d$, and $m + 37d$ merging the data from all the provincial delegations.
- Post-collection data editing is conducted upon these batches.
- Computation of final indices and variation rates is conducted in each batch.
- Press release takes place at $m + 51d$.

The proposed early estimates of the indices aim at producing the same output as the press release as soon as data are available for processing at Statistics Spain's central office (with the exception of the breakdown per market and the seasonal and calendar-adjusted indices).

The aim of this project is to impute, daily, using the data of the responding units, the whole sample of the Industrial Turnover Index (ITI) using Statistical Learning Algorithms. The final objective is to calculate a daily index for the dissemination plan using both the data from the questionnaires received and the imputations of the non-respondents calculated with ML.

2.3.2. Use of Machine Learning

The expected value added from ML is the imputation of the data of the missing units. In comparison with the use of traditional methods of imputation, we expect from ML more accurate predictions as these techniques can use all the available information (longitudinal, current, external) in a more flexible way, detecting changing patterns. One model is trained for all the sampling units each month and it is used to predict the data of non-response unit daily with the new information collected.

2.3.3. Exploratory Data Analysis

Description of the input data:
The input data are:

- Historical monthly data, since 2015, for each unit in the ITI sample. These data are not only edited, but also validated by the survey managers, and used for dissemination. The variables included are the identification variables (VAT number; enterprise and establishment CNAE class and postal code; NUTS3 and municipality code) and the target variable (monthly turnover).
- Edited data of the reference period for the ITI sample. These data are available daily for the units that have answered the questionnaire.

Since the first periods are needed for the construction of regressors and the initial training data, the series of results comprise 68 consecutive months from May 2016 to December 2021. For each reference month we compute five values, namely the initial prediction without current data, the early estimates for the three batches, and the final validated value. The early estimates are computed together with their conditional root mean squared error. We have reconstituted the 7 types of breakdown for this index for each of their respective categories (see table 2.1).

Breakdown	**No. Categories**
General	1
NUTS2	17
MIGS	5
MIGS2	4
NACE Rev. 2 Section	2
NACE Rev. 2 Division	28
NACE Rev. 2 Division-Group[3]	38
Total	95

Table 2.1 Number of categories per index breakdown.

Data Analysis:

The evolution of units not yet collected (missing) per batch in each period is very similar as it is shown in Figure 2.3. This figure shows the proportion of missing units per batch for each month of 2021. A clear outlier is observed in July 2021, which maintains a significantly higher volume of missing data at batch 29. This is primarily because July data are collected during August, when summer vacations lead to slower response rates and a higher reliance on early imputation during these intermediate stages. The specific values are shown in a table in appendix B in detail.

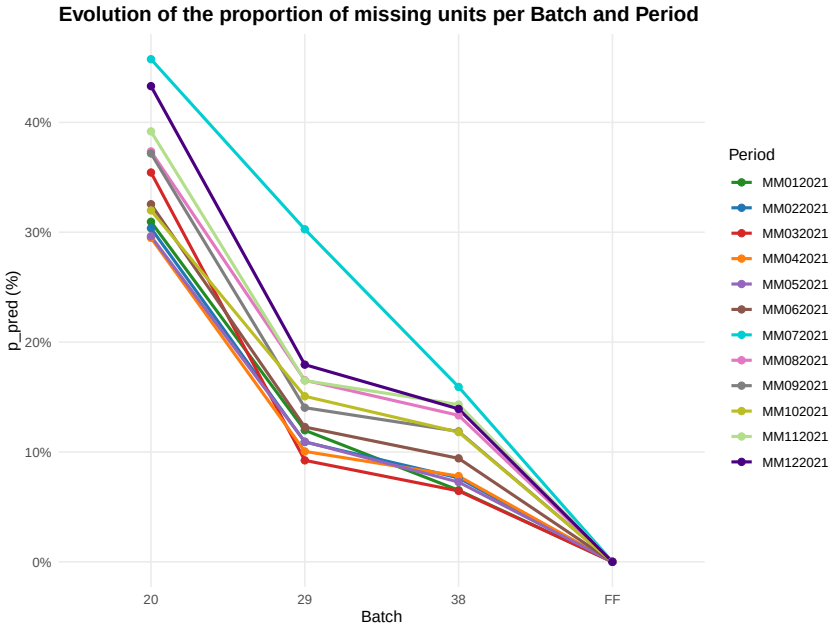

Figure 2.3 Evolution of the proportion of units to be predicted from January to December, 2021.

2.4. Early imputation in accommodation establishments (PL) [PL-Accommodation]

2.4.1. Description

The aim of this project is to address challenges in producing reliable statistics for accommodation establishments. The focus is on developing flash estimates based on data collected with varying levels of timeliness. The ultimate goal is to improve the quality of data imputation models for accommodation establishments.

The analyzed data come from the KT1 survey ("Report on the occupancy of tourist accommodation establishments"). The survey covers tourist accommodation facilities regardless of their type, ownership, or location, as well as other facilities not primarily intended for tourism but temporarily used by tourists (e.g. dormitories or sports and recreation centers). Within the scope of this project, only the total number of tourists will be analyzed. The survey includes only establishments with at least 10 available beds.

Reports are expected to be submitted by the 10th day of the month following the reporting month; in practice, however, submissions occur throughout the month. Some entities fail to submit reports, or submit them late. The response rate is around 80%, but it varies depending on the season. This variable level of data completeness is one of the main reasons why early estimates and data imputation are necessary for this survey.

2.4.2. Use of Machine Learning

On micro data level Machine Learning is capable to ensure admissible results compared to standard techniques e.g. linear regression.

Quality of results will be examined based on range of metrics relevant to numerical data by comparison of flash estimates for a given point in time to the final results. Standard methods e.g. k-fold cross-validation shall be applied.

2.4.3. Exploratory Data Analysis

Description of the input data:
In the initial phase of the project, it was planned to acquire additional data through web scraping. Integrating scraped data with survey data appeared promising; however, we encountered some issues with perfectly matching the two data sources. At this stage, we fully rely on survey data.

Data Analysis:
Total number of tourists has a strong seasonal pattern (Figure 2.4) with a peak during summer holiday.

Harsh decrease was observed during COVID-19 pandemic. Thus, it needs to be model to take into account that decrease was abnormal. Spatial heterogenity (Figure 2.5) results mainly from the availability of tourist attractions, and the natural features of the regions, e.g. mountains (southern region), lakes, sea (north-west region), etc. as well as due to business purposes (capital region).

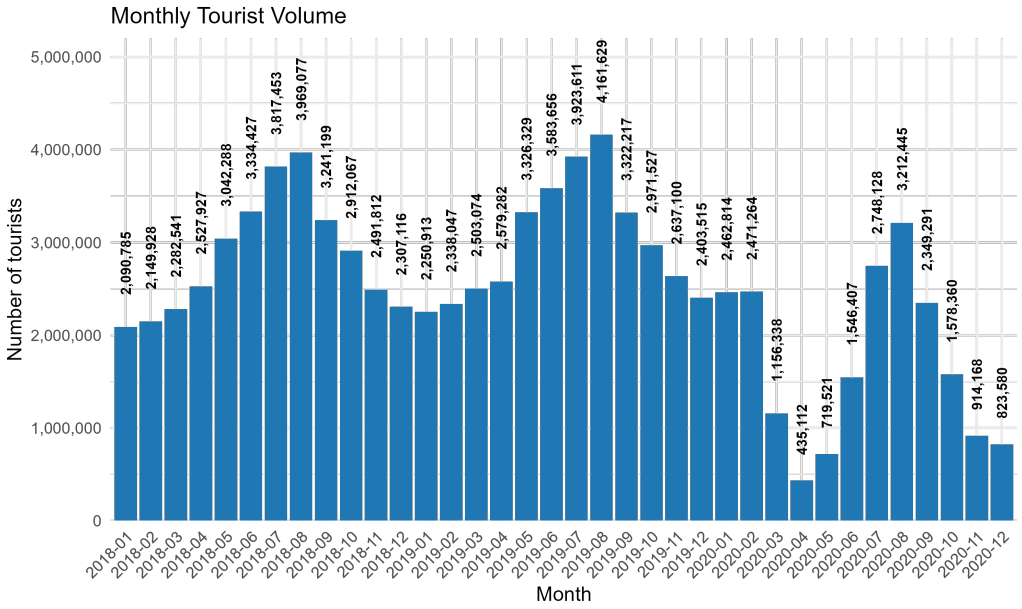

Figure 2.4 Monthly number of tourists

Some significant differences may be observed between types of accommodation establishments, e.g. higher number of tourists in hotels on average than in apartaments.

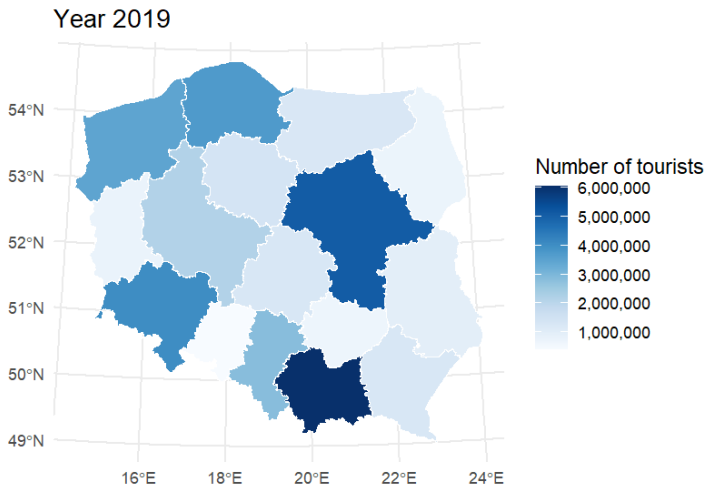

Figure 2.5 Spatial ditribution of tourists

3

Data preprocessing

Data preprocessing is the first step in the ML imputation process. It involves cleaning, transforming and organizing raw data to ensure it is accurate, consistent and ready for its subsequent use. In this chapter we will analise the different steps in the data preprocessing phase, provinding some examples of each project. Data preprocessing has a big impact in the results, therefore we should be very careful at this step.

3.1. Data preparation

Data preparation is a critical first step in any machine learning pipeline, as the quality of the data directly impacts model performance [Kuhn and Johnson, 2013]. This phase involves transforming raw data into a clean and usable format, addressing issues that could otherwise lead to biased or inaccurate models. The main objectives are to handle missing values in the regressors, correct inconsistencies, and normalize the data scale.

3.1.1. Treatment of missing values

Real-world datasets often contain missing values due to collection errors, system failures, or voluntary omissions. These gaps must be addressed before modeling, as most algorithms cannot handle them directly. In this step the missing values to be treated are located in the regressors variables with the goal to complete the initial dataset. This imputation in the regressors is done both in the training dataset and in the predict dataset where the target is missing so predicted by the ML models.

Several imputation strategies exist that can be used for the missing in the regressors:

- **Mean/Median/Mode Imputation**: Replacing missing numerical values with the column mean or median, and categorical values with the mode. While simple, this method can reduce variance and ignore relationships between variables [James et al., 2013].
- **K-Nearest Neighbors (KNN) Imputation**: Estimating missing values based on the k most similar observations. This approach preserves relationships but is computationally expensive for large datasets [Troyanskaya et al., 2001].
- **Model-Based Imputation**: Using predictive models (e.g., regression, decision trees) to estimate missing values iteratively. Advanced techniques like Multiple Imputation by Chained Equations (MICE) account for uncertainty by creating several imputed datasets. In contrast to simple mean imputation, the flexible imputation methods described by Van Buuren [2018] ensure that the multivariate relationships and the inherent uncertainty of the missing values are preserved in the final analysis.

The choice of method depends on the data's nature, the amount of missing data, and the underlying assumption about the missingness mechanism (Missing Completely at Random - MCAR, Missing at Random - MAR, or Missing Not at Random - MNAR), see Rubin [1976].

3.1.2. Outlier detection and treatment

Outliers are observations that significantly deviate from the rest of the data and can distort statistical analyses and model parameters. Detection methods for continuous variables include:

- **Statistical Methods**: Using interquartile range (IQR) or Z-scores to identify values beyond typical thresholds. For example, it can be considered as outliers values beyond 1.5*IQR or $|Z| > 3$ assuming symmetry which is often not encountered with enterprise data. Skewness adjusted boxplot can be used, more details in McGill et al. [1978] and Hubert and Vandervieren [2008].
- **Model-Based Methods**: Employing isolation forests or one-class SVMs to detect anomalies in high-dimensional data [Liu et al., 2008].

Treatment options include removal, transformation (e.g., winsorization, log transform), or treating them as a separate category, depending on whether they represent errors or genuine rare events.

3.1.3. Data normalization and standardization

Many algorithms (e.g., SVM, KNN, gradient descent-based models) require features to be on a comparable scale to prevent dominance by variables with larger ranges. Common techniques include:

- **Min-Max Scaling**: Transforms values to a fixed range, typically [0, 1]: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$.
- **Standardization (Z-score Normalization)**: Centers data around zero with unit variance: $x' = \frac{x - \mu}{\sigma}$, where μ is the mean and σ the standard deviation. This is less affected by outliers [Pedregosa et al., 2011]. In case of problems with skewed data see Rousseeuw and Croux [1993].

3.1.4. Practical information from the projects

3.1.4.1. Early imputation in school enrollment (IT)

The first step of exploratory data analysis aims to understand the structure of the available data through descriptive univariate and multivariate analyses, examining the relationships between the available input features and the output variable. In particular, the association between the response variable and the covariates was preliminarily investigated by comparing the distribution of school enrollment across the different categories of the potential covariates.

The results can help identify the characteristics that best explain school enrollment, potentially suggesting alternative feature encodings or the creation of new variables based on interactions among the observed ones. In particular, the following flag variables are derived after the analysis on the dataset 1 (on primary and lower secondary school enrolment):

- `fl_ita`: Citizenship, coded as 0 (not Italian) or 1 (Italian)
- `fl_gt14`: Identifier for individuals older than 14 years (in a standard educational pathway, lower secondary school is typically completed at age 14)

- `fl_M3`: Identifier for individuals enrolled in the final year of lower secondary school (3rd year)
- `fl_fail`: Identifier for individuals who have been failed in the past (derived from the variables year of attendance)
- `fl_IntPariM3`: Identifier for individuals attending the final year of lower secondary school (3rd year) in a paritarian school.
- `fl_mun`: Identifier for individuals enrolled in a school located in the same municipality as their residence (derived from the variables municipality of residence and municipality of school)
- `fl_ch`: Identifier for individuals who have changed schools from one year to the next (derived from the school identification code).

These variables were used to estimate school enrollment through a logistic regression model, as the original variables could not be included due to their excessively large number of categories. For the final selection of covariates to be included in the logistic regression model, a stepwise selection was applied.

The exploratory analysis confirms that information on demographic characteristics is complete, while some missing values occur in the longitudinal administrative data. Missing values are imputed with predefined values. Specifically, three cases occur:

- school characteristics (municipality and type of primary or lower secondary school) are unavailable for a small portion of the enrolled population (less then 1% of missing data each year). The solution adopted is as follows:
  - the missing school municipality is imputed using the individual's municipality of residence. This choice is based on the fact that the majority of enrolled individuals (85%) attend a school located in their municipality of residence;
  - the missing type of school is imputed with the value 1 (State school), which represents the modal value of the variable and accounts for approximately 95% of the observed cases.
- all information related to the school is unavailable for individuals who are not enrolled. This becomes a problem if we decide to include all available past information as features in the model. Since these missing values are structurally missing, they are encoded with a predefined out-of-range value.

3.1.4.2. Early imputation in the industrial turnover index in Germany (DE)

For estimates produced at time $t + 15$, the share of data available in edited form is relatively low. Therefore, an outlier detection procedure is applied to the remaining unedited values in order to determine whether they should be treated as valid observations or imputed. This is done by comparing the month-on-month change in turnover of a local KAU to the corresponding development within its sector at the four-digit level. Based on this comparison, an acceptable range is defined. Values falling within this range are treated as sufficiently reliable and are not imputed, whereas values outside the range are considered missing and are subsequently imputed.

Regarding the treatment of missing values in the predictors, two alternative approaches are considered: a one-step and a two-step method. In the one-step approach, all missing values—both in the regressors and in the target variable—are estimated simultaneously within

an iterative procedure. In contrast, the two-step approach first imputes missing values in the regressors, for example using linear regression, before imputing the target variable in a second step. A drawback of this sequential procedure is that imputed regressor values are treated as observed in the subsequent model, which may lead to an underestimation of the uncertainty associated with the imputation process (see also Yadegar et al., 2025).

3.1.4.3. Early imputation in the industrial turnover index in Spain (ES)

For example in the case of ES-ITI, they resort to the nested structure of official statistical classifications to propose the general imputing rule for any missing value in numerical regressors: any missing value in a numerical regressor is imputed by the mean of the same regressor in the immediately hierarchically superior category. For example, if regressor `mean_cnae3_est` (mean of establishment turnover by NACE Rev. 2 group) is missing, we impute it with the value `mean_cnae2_est` (mean of establishment turnover by NACE Rev. 2 division). This general rule is complemented with a similar rule for regressors computed with past values. For example, missing values in the regressor `quantile_MA12subdiv_3` (value of the 12-month moving average ecdf $F^*_{\text{MA12subdiv}}$ at validated value $z_k^{m-3y,\text{val}}$) are imputed by `quantile_MA12subdiv_1` (the immediately most recent computed value). This is also applied to geographical categories. This general rule is indeed applied in real production conditions. In remaining highly unlikely missing values in categorical variables, they would be treated as a category itself.

It is important to keep in mind that industrial turnover data have non-negligible representative outliers [Chambers, 1986]. These are readily explained by the standard industrial activity of some notably large firms. As a matter of fact, there exist industrial establishments with a strong influence on both the released indices and annual variation rates. It is essential, though not easy, to be able to predict these values with some degree of accuracy. To this end, for the time being we shall not process them in a special way but we shall use this fact as an orientation to make an adequate choice of regressors and the prediction model.

3.1.4.4. Early imputation in accommodation establishments (PL)

The database for modelling purpose was derived by combining three annual databases for the period 2018-2020, where each annual database consisted of monthly reports of accommodation establishments. Due to seasonal activity of accommodation establishments, some reports were missing for the period of inactivity. The database was transformed to long format. Hence, every accommodation establishment was represented by several instances covering particular years and month. Several variables where recoded for modelling purposed. Some additional variables were supplemented from survey frame. At this stage, no instances were removed.

3.2. Feature engineering

Feature engineering is the process of creating new predictors or transforming existing ones to improve model performance. It requires domain knowledge and creativity, as well-designed features can often yield greater performance gains than algorithm selection alone [Domingos, 2012].

In terms of the representation learning hierarchy, Figure 3.1 illustrates the evolution of approaches in artificial intelligence, contrasting traditional methods with modern deep learning techniques. The diagram presents three distinct paradigms for connecting input data to desired outputs.

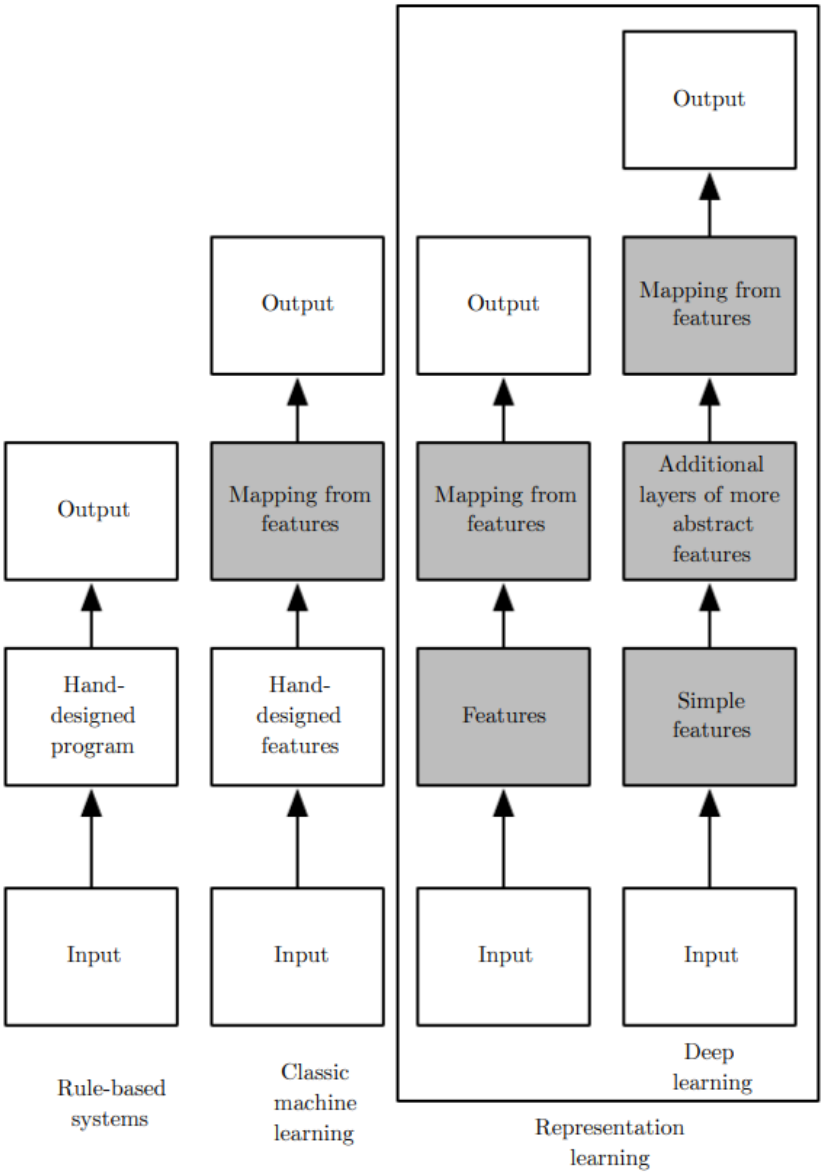

Figure 3.1 Flowchart of how different AI approaches connect representations to outputs [Goodfellow et al., 2016].

In the leftmost path, **rule-based systems** represent the earliest AI approach, where human experts manually design programs with explicit logical rules to process inputs and produce outputs. While interpretable, these systems lack flexibility and cannot handle complex, unstructured data.

The middle path depicts **classical machine learning**, which introduced a crucial advancement: instead of programming rules directly, engineers design feature extractors that transform raw input into informative representations. These handcrafted features (such as HOG for images or TF-IDF for text) are then fed to statistical models that learn the mapping to outputs. This approach reduced the need for explicit rule programming but still required substantial domain expertise for feature engineering.

The rightmost path showcases **deep learning**, the most automated approach. Here, the model learns both the feature representations and the final mapping simultaneously through multiple layers of abstraction. Starting from raw input, the network autonomously discovers simple features (edges, textures), combines them into more complex patterns (object parts),

and finally assembles these into high-level concepts for output generation. This end-to-end learning eliminates the need for manual feature engineering and has demonstrated superior performance on complex tasks like image recognition, natural language processing, and game playing [Goodfellow et al., 2016].

The vertical progression in the diagram highlights increasing automation in representation learning, with deep learning pushing automation furthest by discovering hierarchical feature representations directly from data. This paradigm shift explains why deep learning has become dominant in handling high-dimensional, unstructured data where manual feature design is impractical.

3.2.1. Feature construction

The information utilized in the construction of new features can be classified according to three fundamental dimensions: temporality, source, and the level of granularity.

- **By Temporal Dimension:**
  - *Longitudinal Information:* Data collected from the same subjects or units repeatedly over time, allowing the model to capture trends and temporal dependencies.
  - *Cross-sectional Information:* Data collected at a single point in time, providing a "snapshot" of the population variables.
  - *Combined Longitudinal and Cross-sectional:* A hybrid approach (often referred to as panel data) that incorporates both temporal changes and individual differences.
- **By Data Source:**
  - *Internal Source Information:* Data originated from the same primary dataset or survey being analyzed.
  - *Auxiliary Information from External Sources:* Data integrated from secondary sources, such as administrative registers or external census data, to improve the predictive power of the model.
- **By Level of Aggregation:**
  - *Unit-level Microdata:* Highly granular information at the individual level (e.g., data per company or per citizen).
  - *Group-level Microdata:* Information categorized by specific cohorts or clusters (e.g., by economic sector, age group, or geographic region).
  - *Aggregate-level Information:* Data summarized at a macro level, such as national totals or global averages, where individual variations are obscured.

Creating meaningful features from raw data can uncover hidden patterns. Common techniques include:

- **Interaction Terms**: Multiplying or combining existing features (e.g., $x_1 \times x_2$) to capture synergistic effects that individual features might miss. This is particularly important in linear models that assume additivity [Hastie et al., 2009].
- **Polynomial Features**: Adding squared or cubic terms (e.g., x^2, x^3) to model non-linear relationships. Care must be taken to avoid overfitting and multicollinearity.

- **Date/Time Decomposition**: Extracting components like day-of-week, month, hour, or is_weekend from timestamp data to capture temporal patterns [Kanaujia and Yadav, 2015].
- **Domain-Specific Features**: Creating ratios, rates, or aggregated statistics based on expert knowledge. For example, in finance, creating price-to-earnings ratios; in retail, creating average purchase per customer.

3.2.2. Feature transformation

In traditional methods, such as regression, transforming features can make their relationship with the target variable more linear or satisfy model assumptions. In the case of ML models is not needed but it could be of help for some algorithms to do some kind of transformation of the data, some possibilities are the following.

- **Logarithmic Transformation**: $x' = \log(x + 1)$, useful for right-skewed data to reduce the influence of extreme values and stabilize variance.
- **Box-Cox Transformation**: A parametric family of power transformations that finds the optimal transformation to achieve normality Box and Cox [1964]: $x'(\lambda) = \frac{x^\lambda - 1}{\lambda}$ for $\lambda \neq 0$.
- **Binning/Discretization**: Converting continuous variables into categorical bins (e.g., age groups, income brackets) to capture non-linear effects or reduce noise.

3.2.3. Feature selection

Not all features contribute equally to predictive power; some may be redundant or irrelevant. Feature selection reduces dimensionality, decreases training time, and can improve generalization by reducing overfitting [Guyon and Elisseeff, 2003]. Methods include:

- **Filter Methods**: Selecting features based on statistical tests in the case that the nature of the data allows to do it (e.g., correlation with target, mutual information, chi-square) independent of the model. It is a fast method, but may ignore feature interactions.
- **Wrapper Methods**: Using the model's performance as an evaluation criterion (e.g., forward selection, backward elimination, recursive feature elimination). It is more computationally intensive but consider feature interactions [Kohavi and John, 1997].
- **Embedded Methods**: Performing selection as part of the model training process (e.g., Lasso regression $L1$ penalty, tree-based feature importance). It is efficient and model-specific [Tibshirani, 1996].
- **Hybrid Methods**: Combination from the previous methods.

Regularization techniques like Lasso (L1) and Ridge (L2) regression can also be viewed as continuous feature selection or shrinkage methods that penalize model complexity [Hastie et al., 2009].

3.2.4. Transfer Learning

According to Pan and Yang [2010], Transfer Learning (TL) aims to improve the learning of the target predictive function $f_T(\cdot)$ in a target domain $\mathcal{D}_T$ using the knowledge in a source domain $\mathcal{D}_S$ and a source task $\mathcal{T}_S$.

Notations and Definitions

A **domain** $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$.

$$\mathcal{D} = \{\mathcal{X}, P(X)\} \tag{3.1}$$

Given a specific domain, a **task** $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$, which can be interpreted from a probabilistic viewpoint as $P(y|x)$.

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\} = \{\mathcal{Y}, P(y|x)\} \tag{3.2}$$

Transfer learning occurs when the target domain and task are different from the source ones ($\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$). This implies differences in feature spaces, marginal distributions, label spaces, or conditional distributions.

3.2.4.1. Transfer Learning for Regression Tasks

In regression problems, where the label space $\mathcal{Y}$ is continuous, information can be transferred through four main approaches according to the "what to transfer" categorization:

Instance-transfer

This approach assumes that although the source data cannot be reused entirely, certain parts of the data in $\mathcal{D}_S$ can be reused with a few labeled data in $\mathcal{D}_T$ by re-weighting. For a regressor, this is often implemented via *importance sampling*. The optimal parameters θ^* for the target regressor are learned by minimizing:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n_S} \frac{P(X_T)}{P(X_S)} l(y_{S_i}, f(x_{S_i}, \theta)) \tag{3.3}$$

where $\frac{P(X_T)}{P(X_S)}$ acts as a weight to correct the covariate shift between domains.

Feature-representation-transfer

This method aims to find a "good" feature representation that minimizes the domain divergence. In regression, this involves learning a transformation U that maps high-dimensional data to a shared low-dimensional space. The regressor then learns to map these shared features to the continuous output y. This is particularly effective in Deep Learning, where early layers of a regressor act as feature extractors shared across domains.

Parameter-transfer

This approach assumes that the source and target regression models share some parameters or prior distributions of hyper-parameters. For example, in a linear regression setting, the model parameters w can be decomposed as:

$$w_S = w_0 + v_S, \quad w_T = w_0 + v_T \tag{3.4}$$

where w_0 represents the shared knowledge between regressors, while v_S and v_T are task-specific parameters. By learning w_0 from the source, the target regressor requires significantly less data to converge.

Relational-knowledge-transfer

In cases where data is not independent and identically distributed and is characterized by multiple relations, knowledge is transferred by mapping the logical connections between entities. For regression, if a model learns the relationship strength between variables in a source domain (e.g., house price factors in City A), these relational dependencies can be used as a structural prior for a regressor in City B.

3.2.5. Practical information from the projects

All the regressors used in each project with a proper explanation is detailed in the tables of the appendix A. Some explanations about the use of these techniques are explained in the following.

3.2.5.1. Early imputation in school enrollment (IT)

The dataset for the experiment includes various types of variables. Apart from age, all other variables are categorical. A process of variable selection and transformation is necessary. When dealing with categorical variables, it's essential to apply appropriate encoding techniques to convert them into a format suitable for ML models.

In our case study, the data to be imputed must be predicted using features corresponding to different time intervals. Specifically, these include:

- Time-independent variables, such as demographic characteristics. Citizenship is considered stable, so only the information for year t is used.
- Time-dependent variables, which change over subsequent years and refer to a period spanning two years, such as the academic year. For primary and secondary school, we only use the most recent information on school enrollment. However, for more advanced educational levels, past data should also be incorporated, with a separate variable included in the model for each distinct reference period. The hypothesis assumed for the estimation of the output variable is that the conditional probabilities of being enrolled in school year t-1/t are the same as the conditional probabilities of being enrolled in school year t/t+1 (time invariance hypothesis). Since the reference year is indicated in the names of the time-dependent variables (e.g. `type21`, `age21`, ...), during the prediction phase, the names in the test set should be modified to match those used for the corresponding variables in the training set (e.g. `type22` should be renamed to `type21` or viceversa).

In relation to feature selection, most of the information in the dataset (excluding those with a large number of categories such as the municipalities and the school identification code) was initially evaluated using a stepwise selection in logistic regression. The logistic regression results, stratified by region of residence, indicate the following variables as significant: province of residence, gender, type of attendance, type of school, specially constructed flag variables (`fl_ita`, `fl_gt14`, `fl_M3`, `fl_fail`, `fl_IntPariM3`, `fl_mun`, `fl_ch`). In the experimentation with RF, two different subsets of features are tested:

1. Subset 1: the same variables used in logistic regression are considered as input data. This approach enables a direct comparison of the performance between methods.
2. Subset 2: to expoit the potential of ML methods, all the original variable in the dataset are considered, excluding the municipalities and the school identification code because of the too large number of categories. This allows also for testing the importance of data preprocessing. The set of features includes:

- demographic characteristics: `age2021`, `gender`, `citizenship`, `region` and `province of residence`;
- all information on school enrollment for the 2020/2021 school year: `year of attendance`, `type of school`, `type of attendance`, `year of attendance for the 2019/2020 school year`.

Handling of categorical variables can vary depending on the ML algorithm employed. A possible future development of this work is to consider Neural Networks in the model evaluation phase, allowing information on municipalities and school identification codes to be included as input features through embedding layers. These embeddings transform categorical variables into continuous vector representations, enabling the model to capture intricate relationships within the data.

In relation to feature transformation, for some potential input variable in the dataset, a proper transformation is applied. Specifically:

- Dummy encoding for cathegorical variables with a low number of cathegories: type of attendance and region of residence;
- Label encoding for variables with a high number of cathegories, were the proximity among cathegories is meaningful: region of residence, province of residence , year of attendance (for school year 2020/2021 and 2019/2020).

In the Random Forest experiments, the effect of different encoding strategies was evaluated for the input variable "Region of residence". Since Python libraries such as `scikit-learn`, which were used in this study, do not natively handle categorical variables, qualitative predictors must be transformed into numeric representations. Accordingly, "Region of residence" was encoded using both dummy encoding and label encoding. Although label encoding may introduce an artificial ordering among categories, the results obtained with the two encoding strategies were highly similar, suggesting that, for this variable, the choice of encoding method has a limited impact on model performance. In contrast, several packages in `R` can handle categorical predictors directly, eliminating the need for explicit numeric encoding.

3.2.5.2. Early imputation in the industrial turnover index in Germany (DE)

Appendix A provides an overview of the features used in the model.

The regressors are selected based on theoretical considerations and empirical evidence to capture the underlying relationships as accurately as possible. Each regressor represents a specific influencing factor whose effect on the dependent variable is quantified within the model.

The imputation model incorporates both temporal and structural influencing factors in order to adequately represent the dynamics of the target variables. Temporal components play a central role, particularly through the inclusion of lagged values such as the preceding month's and the preceding year's turnover. These variables capture short-term developments as well as longer-term trends and seasonal patterns. In addition, moving averages for turnover over periods of three and twelve months are used to smooth out short-term fluctuations and to identify more stable patterns in turnover development.

Beyond turnover history, further economic indicators are taken into account. For example, a six-month moving average of new orders is included as an early indicator of future turnover. Price developments are incorporated by considering the previous month's domestic price level at the four-digit industry level, allowing the model to reflect the impact of price changes on turnover. Structural differences between local units are captured through categorical variables such as the type of unit (single- or multi-unit enterprises) and the German federal states.

These variables allow for differentiation between local units with respect to business structure and regional economic conditions. Calendar effects, including the number and distribution of working days, public holidays, and seasonal patterns, are also considered. In addition, the number of persons employed in the preceding month is included as a scaled operational indicator reflecting production capacity and potential changes in turnover. To capture systematic temporal effects and long-term developments, time-fixed effects for each reference year and month are incorporated.

Overall, this comprehensive set of regressors enables the imputation of missing values while accounting for both economic interdependencies and structural characteristics across sectors and regions. This allows for precise estimation of turnover developments at the level of local KAUs and provides a reliable basis for further analysis. However, a key challenge remains: the values reported early in the current reference month are not fully representative of the population, as larger and more heterogeneous local units tend to report later. Consequently, these early-reported values are not suitable as regressors for optimally estimating missing values for the respective month (see also Yadegar et al., 2025).

3.2.5.3. Early imputation in the industrial turnover index in Spain (ES)

Regarding the regressors, we have defined variables reflecting the outlying behaviour of turnover values. In particular, we have defined regressors such as 95th percentiles, indicator values thereof, maximum values per domain, values of empirical cumulative distribution functions, etc. (see appendix A). In this way, the outlying behaviour in the target variable values in preceding time periods is also contained in multiple regressors.

Primary regressors are taken directly from the collection, editing, and estimation paradata, microdata, and aggregates of each survey. In turn, derived regressors are computed for each batch and each reference time period following their metadata specifications for all units in the cut-off population.

We underline the fundamental idea that regressors are chosen so that the prediction model can be applied to units both in the subset of respondent units at batch of day d, $k \in r(d)$, and in the not yet collected units, $k \in S - r(d)$, thus discarding regressors at the unit level for the current reference time period. Instead, only aggregated quantities such as quantiles, means, standard deviations, etc. are used as regressors with reference to this time period.

Notice also that we can build a preliminary prediction model in which regressors from the current reference time period are not included. This will allow us to predict turnover values before any data from the current reference time period is collected (a genuine prediction exercise) at $m + 1d$. Thus, we shall be able to assess the relevance of data from the current reference time period, even from a fraction of the sample, to produce a reliable early estimate. Detailed results about this exercise are shown in the chapter of results.

As stated, there exist two types of units, namely, those included in the corresponding batch $k \in r(d)$ and those not $k \in S - r(d)$. For those regressors whose computation involves data from the current reference time period (i.e. value 0 is contained in the regressor metadata attribute named Time Periods). These regressors are aggregated variables, so that we can straightforwardly construct these by the corresponding value for the cross-domain variable(s) specified in the metadata.

For example, the regressor `mean_trnovr_ed_NACE2div` is computed as the mean of the edited turnover values by NACE Rev.2 division using only units in $r_D(d)$, the sample of

respondents in the domain D at time d (complete-cases analysis according to Little and Rubin [2002]):

$$\bar{z}_D = \frac{1}{N_{r_D(d)}} \sum_{k \in r_D(d)} z_k^{\text{ed}}, \qquad D \in \text{NACE2div}.$$

For those units $k \in S_D - r_D(d)$ (thus not yet with a value for the edited turnover), the value is constructed as $x_k(d) = \bar{z}_D(d)$, which is the value for the corresponding domain U_D. Note that when we use NACE Rev. 2 refers here to the Spanish National Classification of Economic Activities (CNAE-2009), adapted from the international NACE Rev. 2.

All regressors using lag 0, i.e. the data from the reference period, are named as *current regressors*. The information about the reference period in these current regressors is transfered from the collected units at the moment of the prediction to the predicted units. The relation between these regressors and the longitudinal regressors is also essential to improve accuracy in the prediction values. There is a transfer learning from the information in the collected units of the reference period to the not yet collected units that will be predicted by the model.

3.2.5.4. Early imputation in accommodation establishments (PL)

Within feature construction stage, utilizing lagged target variable, NUTS and LAU identifiers, type of accommodation establishment, information about COVID-19 restriction, in total 62 variables were constructed. For time window of 3, 6 and 12 months, several statistics were calculated, namely:

1) mean,
2) median,
3) the first quartile,
4) the third quartile,
5) sum,

with respect to NUTS and LAU identifiers, type of accommodation establishment. Also lags of order 1, 2 and 3 of target variable were included.

During the feature selection process, the following methods were used: analysis of the correlation matrix of available features, Hellwig's information capacity indices. Additionally, as an aternative, stepwise regression was applied in three modes: forward stepwise selection (starting from the null model), backward elimination (starting from the full model) and bidirectional elimination. The Lasso method was also used due to its parsimony (lasso model explains the data with as few variables as necessary).

Since there were over 60 explanatory variables, the top 20 explanatory variables were selected with respect to Pearson correlation coefficient with the target variable, as a starting point. Top 20 variables are presented in Figure 3.2.

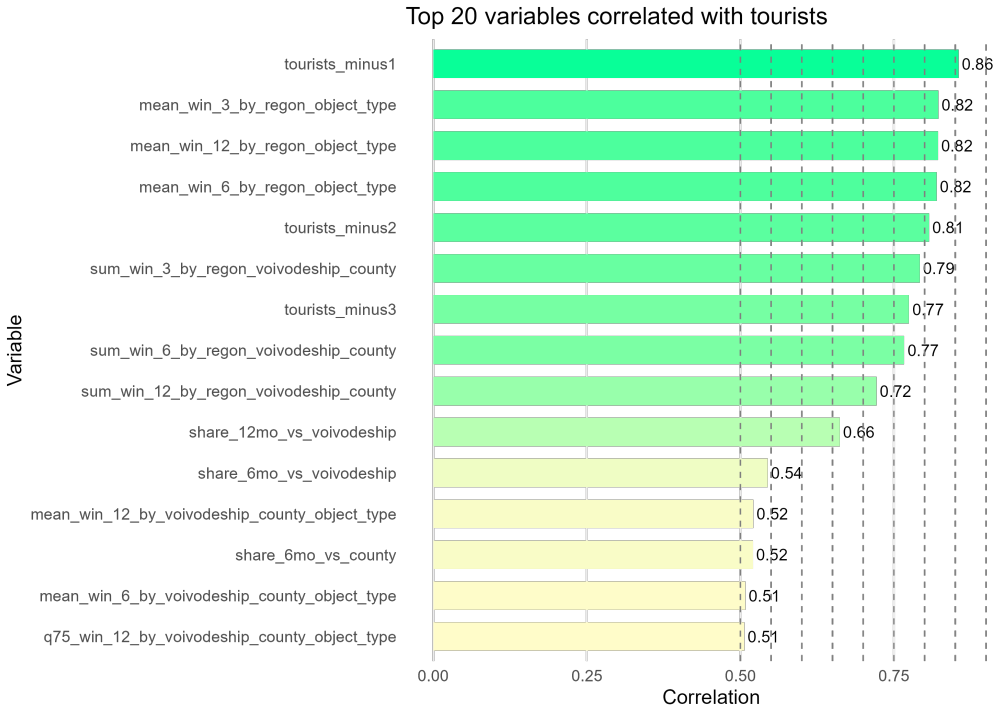

Figure 3.2 Pearson correlation coefficient for the top 20 variables

Hellwig's information capacity indices method was applied for these 20 variables. Hellwig's method is a quantitative procedure designed for the selection of explanatory variables in statistical and econometric models. The method was proposed by Polish statistician Zdzisław Hellwig (1968) and is based on the concept of the *information capacity index* of a set of explanatory variables. Its objective is to identify the subset of predictors that provides the greatest amount of information about the dependent variable while limiting redundancy among regressors.

The key idea of Hellwig's method is to select a combination of explanatory variables that:

- is strongly correlated with the dependent variable,
- exhibits low intercorrelation among explanatory variables,
- maximizes the so-called *integral information capacity*.

Thus, the method simultaneously accounts for explanatory power and multicollinearity.

Assume that Y denotes the dependent variable and $X_1, X_2, \ldots, X_m$ represent the set of potential explanatory variables. First, we define all combinations of potential explanatory variables. For m candidate variables, $2^m - 1$ non-empty subsets (combinations) are possible. Each subset is treated as a separate candidate model. For each variable X_j belonging to the k-th combination, the individual information capacity is computed as:

$$h_{kj} = \frac{r_j^2}{1 + \sum_{\substack{l=1 \\ l \neq j}}^{m_k} |r_{lj}|}$$

where:

- r_j is the Pearson correlation coefficient between X_j and Y,
- r_{lj} is the Pearson correlation coefficient between explanatory variables X_l and X_j,

- m_k is the number of variables in the k-th combination.

The numerator reflects the explanatory power of variable X_j, while the denominator penalizes high correlation with other explanatory variables in the same combination. The total (integral) information capacity of the k-th combination is defined as:

$$H_k = \sum_{j=1}^{m_k} h_{kj}$$

The value H_k represents the overall information content of a given subset of explanatory variables. The optimal combination of explanatory variables is the one that maximizes the integral information capacity:

$$H_k = \max\{H_1, H_2, \ldots, H_{2^m-1}\}$$

Keep in mind that the number of possible combinations grows exponentially with the number of candidate variables, and the method relies on linear correlation coefficients and may not capture nonlinear relationships.

With Hellwig's method, the set of 20 predictors was reduced to 15 (cf. Figure 3.3)

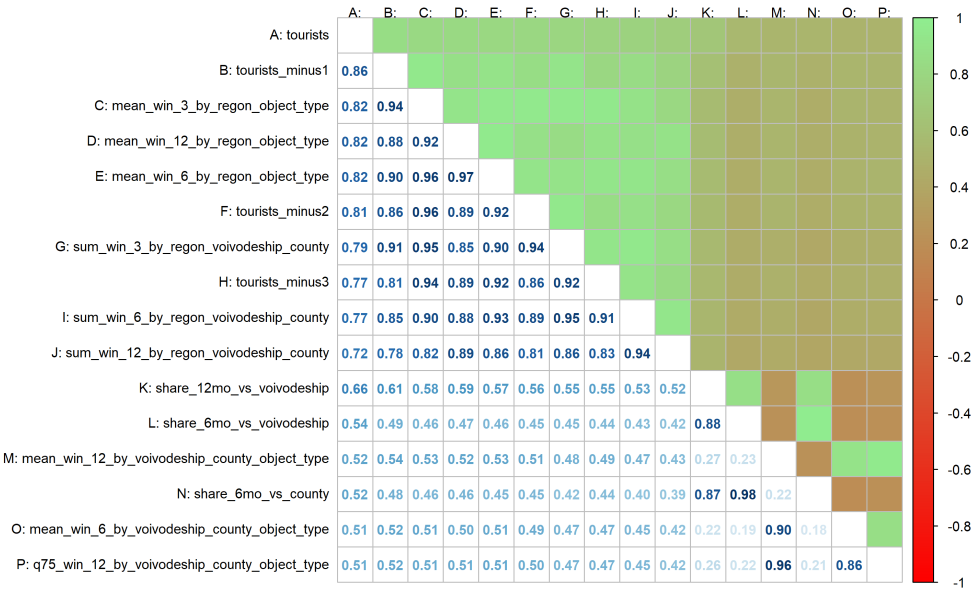

Figure 3.3 Correlation matrix for the top 15 covariates from Hellwig's method

As alternative, two methods were used for comparison: LASSO model and stepwise regression. In a fact, both methods cover feature selection and model parameter estimation. The next table presents LASSO results.

Table 3.1 LASSO results

Variable	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	-58.47	3.348	-17.465	< 0.0001
tourists_minus1	0.5326	0.009284	57.370	< 0.0001
mean_win_12_by_regon_object_type	0.2289	0.009644	23.734	< 0.0001
tourists_minus2	0.05194	0.009424	5.512	< 0.0001
share_12mo_vs_voivodeship	1141.0	27.21	41.931	< 0.0001
mean_win_6_by_county_object_type	0.1112	0.004913	22.625	< 0.0001

LASSO model revealed its parsimony property - with only five features we may explain substantial part of variability of the target variable.

Further, all three modes of stepwise regression lead to the same results, that is 12 significant explanatory variables, as seen in the next table. They cover the set of explanatory variables derived from LASSO model.

Table 3.2 Estimation Results for the LASSO Model

Variable	Estimate	Std. Error	t value	Pr($>\|t\|$)
Intercept	-48.83	3.628	-13.460	< 0.0001
tourists_minus1	0.4885	0.01321	36.969	< 0.0001
share_12mo_vs_voivodeship	1281.0	48.09	26.642	< 0.0001
mean_win_12_by_regon_object_type	0.5361	0.02235	23.985	< 0.0001
mean_win_6_by_county_object_type	0.1798	0.01253	14.348	< 0.0001
sum_win_12_by_regon_county	-0.01460	0.001286	-11.351	< 0.0001
mean_win_12_by_county_object_type	-0.09151	0.01449	-6.313	< 0.0001
tourists_minus2	0.02893	0.01735	1.667	0.09552
mean_win_6_by_regon_object_type	-0.3999	0.03884	-10.296	< 0.0001
sum_win_6_by_regon_county	0.02563	0.005100	5.026	< 0.0001
mean_win_3_by_regon_object_type	0.1691	0.03463	4.885	< 0.0001
share_6mo_vs_voivodeship	-321.5	73.35	-4.384	< 0.0001
share_6mo_vs_county	212.8	69.74	3.052	0.00228

Stepwise Selection

To select the most important variables explaining the number of tourists, stepwise regression was applied in both directions (2D-C method with direction `both`), analyzing both addition and removal of variables based on goodness-of-fit criteria such as AIC.

First, a full model `full_model4` containing all available variables and an empty model `empty_model4` containing only the intercept were prepared. Then, the `step` function in R was used, defining the scope between the empty and full model with direction `both`, to automatically select the combination of variables that best explain the dependent variable `tourists`.

The fitted stepwise model selected 12 variables, including `tourists_minus1`, `tourists_minus2`, various winter average values by region and type of facility, and shares of tourists over 6 and 12 months relative to the province and county. Some variables were rejected if their inclusion did not significantly improve the model fit.

After fitting the model, forecasts were calculated for the entire dataset, and basic error statistics were computed: RMSE = 265.3, MAE = 115.3, MAPE could not be computed due to the presence of zero values in the dependent variable, and the mean error (ME) was practically zero. The model exhibits a high goodness of fit, as indicated by the coefficient of determination $R^2 = 0.8908$.

Stepwise selection proved to be an effective method for automatically choosing relevant variables, providing good interpretability and minimizing the risk of overfitting. This method allows accounting for correlations between variables and selecting those that truly contribute explanatory power, which is especially useful when dealing with a large number of potential predictors.

3.3. Imbalanced data

In the case of categorical target, imbalanced data is a common challenge in machine learning where the number of observations in one class significantly exceeds the others. This occurs frequently in real-world applications such as fraud detection, medical diagnosis, and anomaly detection, where the event of interest is rare. Class imbalance can lead to biased models that favor the majority class, resulting in poor predictive performance for minority classes despite high overall accuracy. To address this,Khan et al. [2024] emphasize the strategic combination of data augmentation and ensemble learning as a primary defense against such bias.

3.3.1. Data-level techniques

Data-level techniques modify the dataset distribution to create a more balanced representation of classes before model training.

Random Undersampling: Random undersampling reduces the number of instances from the majority class by randomly removing samples until class balance is achieved. While computationally efficient, this approach may discard potentially useful information.

```
# Pseudocode for random undersampling
def random_undersampling(X, y, target_ratio=1.0):
majority_class = get_majority_class(y)
minority_class = get_minority_class(y)
X_majority, y_majority = get_class_samples(X, y, majority_class)
X_minority, y_minority = get_class_samples(X, y, minority_class)

n_samples = int(len(X_minority) * target_ratio)
indices = random_select_indices(X_majority, n_samples)

X_balanced = concatenate(X_majority[indices], X_minority)
y_balanced = concatenate(y_majority[indices], y_minority)
return shuffle(X_balanced, y_balanced)
```

Random Oversampling: Random oversampling increases the number of instances in the minority class by randomly duplicating existing samples. This preserves all information from the original dataset but may lead to overfitting due to exact replication of minority class instances.

Synthetic Minority Oversampling Technique (SMOTE): SMOTE [Chawla et al., 2002] generates synthetic samples for the minority class by interpolating between existing instances. For each minority class sample, SMOTE selects k nearest neighbors and creates new samples along the line segments connecting the sample to its neighbors.

$$x_{\text{new}} = x_i + \lambda \times (x_{zi} - x_i) \tag{3.5}$$

where x_i is a minority class instance, x_{zi} is one of its k nearest neighbors, and λ is a random number between 0 and 1.

Adaptive Synthetic Sampling (ADASYN): ADASYN [He et al., 2008] improves upon SMOTE by generating more synthetic samples for minority class instances that are harder to learn. It adaptively shifts the classification decision boundary toward difficult examples by focusing on minority samples that are surrounded by majority class instances.

3.3.2. Algorithm-level techniques

Algorithm-level techniques modify the learning algorithm itself to address class imbalance without changing the data distribution. The most common algorithm-level technique is class weighting that addresses class imbalance by assigning a higher importance to minority-class instances during training. Instead of modifying the dataset, the learning algorithm incorporates class-dependent penalties directly into the loss function.

Each class c is assigned a weight w_c, typically inversely proportional to its frequency in the dataset:

$$w_c = \frac{N}{K \cdot n_c},$$

where N is the total number of samples, n_c is the number of samples belonging to class c, and K is the number of classes. This formulation ensures that minority classes receive higher weights.

Given a standard cross-entropy loss, class weighting modifies the objective function to:

$$\mathcal{L}_{\text{weighted}} = -\sum_{i=1}^{N} w_{y_i} \log p(y_i \mid x_i),$$

where w_{y_i} is the weight corresponding to the true class of instance i, and $p(y_i \mid x_i)$ is the model-predicted probability.

Most machine learning libraries allow direct specification of class weights:

- **Logistic Regression / SVM:** pass `class_weight=`$\{0 : w_0, 1 : w_1\}$
- **Neural Networks:** provide a `class_weight` dictionary to the training routine
- **Tree-Based Models (e.g., XGBoost, LightGBM):** set parameters such as `scale_pos_weight` or `class_weight`

By increasing the cost of misclassifying minority-class samples, the model becomes more sensitive to under-represented classes, reducing bias toward the majority class without requiring oversampling or undersampling.

Some algorithm-Level techniques for imbalanced data are the following:

Cost-Sensitive Learning: Cost-sensitive learning assigns different misclassification costs to different classes. The objective function is modified to penalize errors on minority classes more heavily:

$$\mathcal{L}_{\text{weighted}} = -\sum_{i=1}^{n} w_{y_i} \cdot \log(p(y_i|x_i)) \tag{3.6}$$

where w_{y_i} is the class weight for the true class of sample i.

Threshold Moving: Instead of using the default 0.5 probability threshold for binary classification, threshold moving adjusts the decision threshold to favor minority class predictions. The optimal threshold can be determined using metrics like the Geometric Mean or Youden's J statistic:

$$\text{Threshold}_{\text{optimal}} = \arg\max_{\tau} \sqrt{\text{Sensitivity}(\tau) \times \text{Specificity}(\tau)} \tag{3.7}$$

Ensemble Methods: Ensemble methods combine multiple models to improve performance on imbalanced datasets.

Balanced Random Forest: Balanced Random Forest [Chen et al., 2004] creates an ensemble where each tree is trained on a balanced bootstrap sample obtained through undersampling of the majority class.

EasyEnsemble and BalanceCascade: EasyEnsemble [Liu et al., 2009] uses AdaBoost on multiple balanced subsets created by undersampling the majority class. BalanceCascade extends this approach by removing correctly classified majority class instances in subsequent iterations.

3.3.3. Evaluation metrics for imbalanced data

Traditional accuracy is misleading for imbalanced datasets. Alternative metrics include:

- Precision and Recall: Precision $= \frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$
- F1-Score: $F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- Area Under the Precision-Recall Curve (AUPRC)
- Geometric Mean: $G_{\text{mean}} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$
- Matthews Correlation Coefficient: MCC $= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

3.3.4. Practical considerations

The choice of imbalance handling technique depends on several factors:

- **Dataset size**: Oversampling is preferable for small datasets, while undersampling is more suitable for large datasets.
- **Class imbalance ratio**: Severe imbalance (e.g., 1:1000 much higher than a moderate around 1:100) may require combined approaches.
- **Computational resources**: Synthetic generation techniques like SMOTE are computationally intensive.
- **Model type**: Some algorithms (e.g., tree-based methods) handle imbalance better than others (e.g., SVM without class weights).

A hybrid approach combining data-level and algorithm-level techniques often yields the best results. For example, using SMOTE for moderate balancing followed by cost-sensitive learning typically outperforms either approach alone.
While hybrid and adaptive strategies often provide superior performance, their effectiveness ultimately depends on the quality of the information carried by the predictors. It is important to emphasize that data-level interventions and training-set selection strategies are beneficial only when the available feature space includes variables with sufficient discriminatory power for the minority class. In the absence of informative predictors, resampling techniques primarily modify the marginal distribution of the response variable without introducing additional knowledge. Consequently, these approaches may lead to a decline in overall predictive accuracy and a reduction in model robustness, rather than a genuine improvement in minority class identification.

Evidence from the case study on children's school enrollment provides a clear example of this mechanism for categorical variables. Although the dataset exhibits a marked class imbalance, its overall size is large enough to ensure that all relevant patterns are already represented in the training data. In this context, oversampling strategies did not improve micro-level estimates; on the contrary, they produced a deterioration in macro accuracy. The most substantial gains were instead achieved through careful hyperparameter optimization. In particular, the best performance emerged from deeper trees, up to the maximum depth, with terminal nodes containing a single observation.

This finding departs from common recommendations in the Random Forest literature, where a minimum leaf size is typically suggested in order to secure an adequate presence of units from the rarest classes within each terminal node. In the present application, however, such a constraint mainly translates into a reduction of the maximum attainable depth. The problem is not the proliferation of leaves with no minority cases—since the rare event is supported by a considerable absolute number of observations—but rather the limited discriminatory power of the regressors. Greater depth allows the algorithm to explore the information embedded in the covariates more thoroughly, ultimately yielding superior predictive performance.

4

Model selection process

Model selection is the process of choosing the most appropriate machine learning model among a set of candidates for a specific dataset. In this document the denomination of model refers to a specific algorithm and specific hyperparameters. If the same algorithm is used with different hyperparameters, that is another model. This selection process involves not only selecting the algorithm (e.g., Random Forest vs. Support Vector Machines) but also finding the optimal configuration with the corresponding hyperparameters that allows each model to generalize well to unseen data.

The model selection process constitutes a fundamental pillar in the construction of robust and generalizable machine learning systems. This chapter presents a comprehensive and technically detailed framework that extends beyond conventional approaches, incorporating advanced techniques for algorithm selection, hyperparameter optimization, and multi-level evaluation. We address not only the predictive accuracy at the instance level but also the quality of aggregated statistics derived from model predictions, which are often the primary focus in practical applications such as official statistics, econometrics, and business intelligence.

Nowadays, a wide range of methods exist to generate predictions for imputation in a data-rich environment, ranging from simple polynomial regression models to neural networks, including random forests and boosted regression trees, to name a few. There already exist excellent initiatives to compare different imputation methods based on diverse statistical learning algorithms to assess their performance [Dagdoug et al., 2021] and a definitive recommendation about the models cannot be provided. There does not exist a best model over the rest, that decision depends on the particular application and even on the data. So, the evaluation of the models shown in this chapter is a must.

4.1. Data splitting workflow

In the development of a Machine Learning model, it is a fundamental requirement to evaluate the generalization error. To achieve this without incurring in *Data Leakage*, the original dataset must be partitioned into independent subsets. The most common procedure is the **Hold-out Method** with an additional validation fold [Géron, 2022]. More details about advanced procedures can be found in the next section of this document. To ensure a model does not simply "memorize" the data (overfitting), the available dataset is typically partitioned into three distinct subsets:

- **Subtrain Set:** The subset of the data, used by the algorithm to learn the underlying patterns and adjust its internal parameters.

- **Validation Set:** Used to provide an unbiased evaluation of a model fit on the subtraining dataset while tuning model hyperparameters. It helps in preventing overfitting during the selection phase.
- **Test Set:** A "gold standard" dataset used only once the final model is chosen. It provides an objective measure of how the model will perform in a real-world environment.

The process is typically carried out in two sequential stages (see Figure 4.1):

1. Phase 1: Initial Partition (Hold-out). The total dataset $\mathcal{D}$ is divided into two main parts:
   - **Training Set ($\mathcal{D}_{train}$):** Used to build the model. Usually accounts for 70-80% of the data.
   - **Test Set ($\mathcal{D}_{test}$):** Also known as the *"Vault"*, it is kept hidden from the model during the entire development phase. It is only used once to estimate the final performance.
2. Phase 2: Validation Split. With the aim of selecting the best model different algorithms can be used in the validation step and for each algorithm there is a **hyperparameters** tuning (such as the learning rate, the number of layers, or regularization constants), the $\mathcal{D}_{train}$ is further divided:
   - **Reduced Training Set (Sub-train):** The actual data used by the algorithm to learn the weights (parameters).
   - **Validation Set ($\mathcal{D}_{val}$):** Used to evaluate the model during training. It provides a "proxy" for the test set, allowing the developer to compare different models (and their hyperparameters) and select the best one without contaminating the test set.

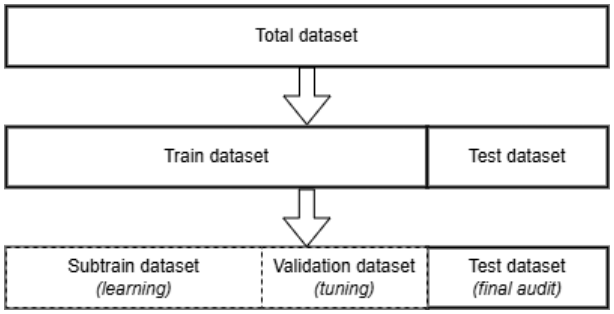

Figure 4.1 Train - Validation - Test

In relation with phase 2 (third line in Figure 4.1), there is no agreement in the literature but from our point of view the tuning step done using the validation split involves both parts: model and hyperparameters with the aim to select the best model.

Mathematical Representation

Let $\mathcal{D}$ be the total set of samples. The partitioning satisfies:

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test} \quad \text{where} \quad \mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset \tag{4.1}$$

Then, the training set is subdivided:

$$\mathcal{D}_{train} = \mathcal{D}_{subtrain} \cup \mathcal{D}_{val} \quad \text{where} \quad \mathcal{D}_{subtrain} \cap \mathcal{D}_{val} = \emptyset \tag{4.2}$$

The **Test Set** must never be used for any decision-making process during the training or hyperparameter tuning phase. Any violation of this rule leads to over-optimistic results and poor generalization in production environments.

The different options to do the split of the train dataset are shown in the next section. Once the tuning of the algorithm and hyperparameters optimization is done, the total dataset is used to train the best model that will be used to predict.

When dealing with categorical predictors in tree-based models, Wright and König [2019] demonstrate that the choice of splitting strategy is crucial for both predictive performance and the prevention of selection bias.

4.1.1. Cross-validation

When the dataset is small, a single validation split might lead to high variance in performance estimates. **Cross-Validation (CV)** addresses this by partitioning the data multiple times.

The most common method is **K-Fold Cross-Validation**, where the data is divided into k subsets (folds). The model is trained on $k-1$ folds and validated on the remaining fold. This process is repeated k times, and the results are averaged. This ensures that every data point is used for both training and validation, leading to a more robust estimation of model performance.

Different ways to split the train dataset in train and validation:

- Random
- Stratified or other methods for data from complex sample survey data, [see e.g. Iparragirre et al., 2023, MacNell et al., 2023, Nalenz et al., 2024, and multiple references therein].
- Cross-validation:
  - K-fold cross-validation
  - Hold-out cross-validation
  - Stratified k-fold cross-validation
  - Leave-p-out cross-validation
  - Leave-one-out cross-validation
  - Monte Carlo (shuffle-split)
  - Temporal cross-validation

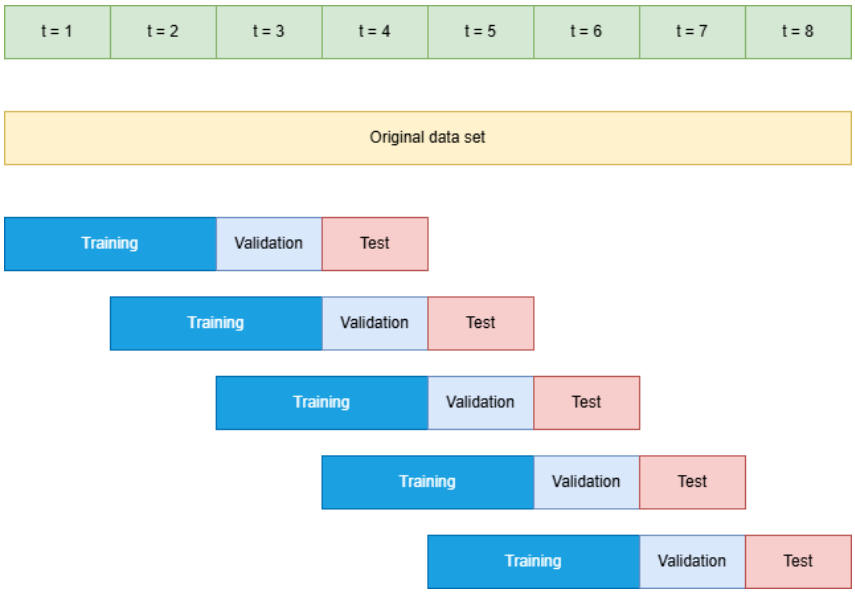

Figure 4.2 Temporal cross-validation with rolling window

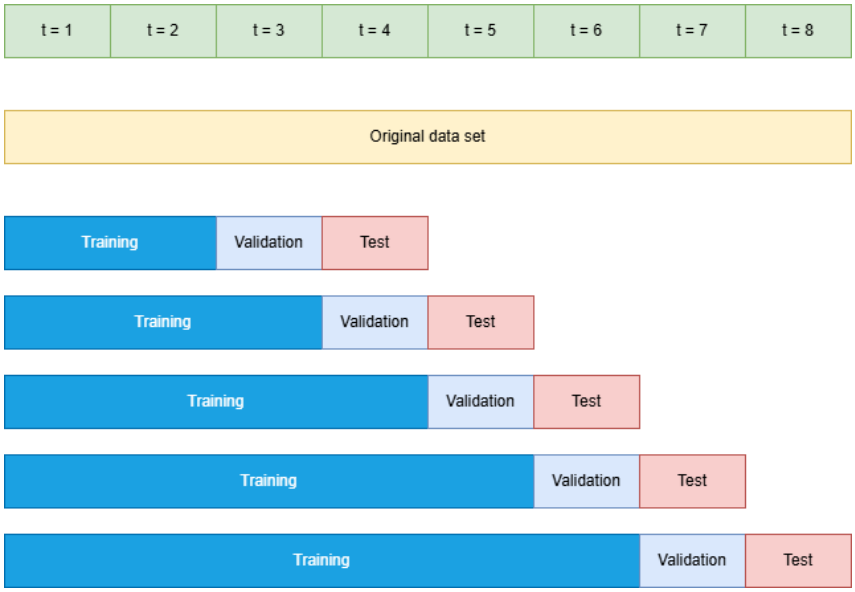

Figure 4.3 Temporal cross-validation with expanding window, also named as hold out

4.1.2. Data leakage

Data leakage occurs when information from outside the training dataset is used to create the model. This leads to overly optimistic performance estimates during validation, which then fail to generalize in production.

There are two primary forms of leakage:

- **Target Leakage:** Including features that are proxies for the target variable or that provide information that would not be available at the time of prediction.
- **Train-Test Contamination:** This happens when the preprocessing steps (like normalization, mean imputation, or feature selection) are applied to the entire dataset before splitting. To avoid this, all transformations must be fitted only on the training set and then applied to the validation/test sets [Kaufman et al., 2012].

To understand this concept simply: it is like giving a student the exam questions while they are studying. They will achieve a 10/10 on the test, but they have not actually learned the subject. In machine learning, this often happens when, for example, the mean of a column is calculated using the whole dataset before splitting; the training set effectively "knows" the distribution of the test set in advance.

The inclusion of leaking features poses significant risks to the model selection process. Specifically, it leads to the selection of a model with poor actual predictive capacity. Because the model becomes overly dependent on the "leaked" information, any minor perturbation or noise in that specific regressor will cause the prediction quality to degrade sharply when the model encounters real-world data.

Detection Strategies

Identifying leakage requires a critical look at the behavior of the model and the relationship between variables. Common detection methods include:

- **Anomalous Feature Importance:** A warning sign is when a single variable exhibits extremely high importance while all other variables show near-zero importance. However, this is not necessarily conclusive, as it may simply reflect a genuinely powerful predictive relationship inherent in the data.

- **Excessive Correlation:** Finding an unusually high correlation between a regressor and the target variable often indicates that the feature contains information that it "should not know" under realistic conditions.
- **Suspicious Performance:** If a model's performance seems "too good to be true" (e.g., near-perfect accuracy on a complex problem), it is often a symptom of underlying leakage.

Some authors include in data leakage also the problem of splitting of complex sample surveys, [Iparragirre et al., 2023].

4.2. Algorithm and hyperparameters optimization

Algorithm optimization focuses on finding the best set of hyperparameters to improve performance. The algorithm selection problem can be formally defined as:

$$a^* = \arg\min_{a \in \mathcal{A}} \mathcal{L}(D_{\text{val}}; a, \theta_a^*)$$

where $\mathcal{A}$ is the set of candidate algorithms, θ_a^* are the optimal hyperparameters for algorithm a, and $\mathcal{L}$ is the loss function on validation data D_{val}.

Optimization in machine learning is a dual-level process. While the algorithm learns the optimal mapping from inputs to outputs, the practitioner must ensure that the configuration of the algorithm itself is optimal for the given task. This section explores the distinction between internal parameters and external hyperparameters, and the methodologies used to navigate the search space efficiently.

4.2.1. Parameters vs. hyperparameters

The distinction between these two concepts is fundamental to understanding how models are trained and tuned:

- **Model Parameters:** These are internal configuration variables that the model estimates from the training data. For instance, in a linear regression, the coefficients (weights) are parameters; in a neural network, the weights and biases are parameters. Their values are determined by minimizing a loss function $\mathcal{L}$ using optimization algorithms like Stochastic Gradient Descent (SGD) [Goodfellow et al., 2016].
- **Hyperparameters:** These are external configurations that cannot be learned directly from the data during the training phase. They govern the learning process itself. Examples include the learning rate α, the number of hidden layers in a network, the regularization parameter λ, or the maximum depth of a decision tree. Hyperparameters must be specified before the training begins [Bergstra and Bengio, 2012].

4.2.2. Hyperparameter tuning

The goal of hyperparameter optimization (HPO) is to find a set of hyperparameters λ^* from a search space Λ that minimizes the generalization error. Formally, if $f(x; \theta, \lambda)$ is a model with parameters θ and hyperparameters λ, we seek:

$$\lambda^* = \arg\min_{\lambda \in \Lambda} \mathbb{E}_{(x,y)\sim\mathcal{P}_{val}}[\mathcal{L}(y, f(x; \theta^*(\lambda), \lambda))] \tag{4.3}$$

where $\theta^*(\lambda)$ are the parameters learned by the algorithm given λ, and $\mathcal{P}_{val}$ is the validation data distribution.

4.2.3. Optimization strategies

The search space for hyperparameters is often high-dimensional and non-convex, requiring systematic search strategies:

- **Grid Search:** Grid search is an exhaustive search through a manually defined subset of the hyperparameter space. While it is simple to implement and parallelize, it suffers from the "curse of dimensionality." If the number of hyperparameters increases, the number of required evaluations grows exponentially. Furthermore, it may waste computational resources exploring dimensions that do not significantly impact the model performance.
- **Random Search:** Random search samples the hyperparameter space based on a defined statistical distribution. Bergstra and Bengio [2012] demonstrated that random search is often more efficient than grid search. This is because, in most cases, only a few hyperparameters are truly "important" for a given dataset. Random search explores more unique values for each dimension, increasing the probability of finding a near-optimal region in fewer iterations.
- **Bayesian Optimization:** Unlike grid or random search, Bayesian optimization is an informed search strategy. It treats the objective function as a "black box" and builds a probabilistic model (a surrogate model, often using Gaussian Processes or Tree-structured Parzen Estimators) to predict the performance of different hyperparameter configurations [Snoek et al., 2012]. By balancing *exploration* (trying areas with high uncertainty) and *exploitation* (probing areas known to perform well), it can find optimal settings with significantly fewer evaluations.
- **Automated Machine Learning (AutoML):** Modern optimization often employs AutoML frameworks that combine the aforementioned techniques with Neural Architecture Search (NAS) or Genetic Algorithms. These methods automate the entire pipeline, from feature selection to the final model ensemble, reducing the need for human intuition in the trial-and-error process.

4.3. Quality evaluation of the model

The evaluation of a machine learning model extends beyond the mere calculation of error metrics. A truly rigorous assessment must consider the integrity of the data used for training and the robustness of the statistical framework used to derive conclusions. This section explores the fundamental principles and advanced methodologies that ensure a model is both reliable and actionable.

Data Integrity: The Garbage In, Garbage Out (GIGO) Principle

The effectiveness of any model selection process is strictly bounded by the quality of the input data. The "Garbage In, Garbage Out" (GIGO) principle serves as a foundational warning in computational modeling: the most sophisticated algorithm cannot compensate for data that is biased, noisy, incomplete, or irrelevant.

In the context of quality evaluation, GIGO implies that a model might exhibit high accuracy on a specific dataset while remaining fundamentally flawed if that data does not represent the real-world phenomenon it intends to capture. Therefore, evaluation must begin with a rigorous assessment of data provenance, feature engineering, and the cleaning process. If the input features are skewed or if data leakage has occurred, the resulting model selection will be an

artifact of flawed data rather than a reflection of true predictive power [Kaufman et al., 2012]. This comprehensive evaluation framework ensures that selected models are not only statistically sound at the individual prediction level but also reliable for producing the aggregated insights that drive real-world decisions. The dual-level assessment bridges the gap between machine learning methodology and practical application requirements.

Total Machine Learning Error (TMLE) model

Puts, Salgado, and Daas [Chapter 2, Section 2.3 of Puts et al. [2025]] develop and propose the Total Machine Learning Error (TMLE) model, establishing a pioneering framework specifically designed to bridge the gap between traditional statistical quality standards and modern algorithmic approaches. The authors argue that standard machine learning metrics—such as accuracy, F1-score, or Mean Squared Error—are insufficient for the rigorous requirements of official statistics, as they focus on model performance rather than the accuracy of the final population estimate. To address this, they adapt the classical Total Survey Error (TSE) paradigm [Groves and Lyberg, 2010] into this new TMLE model, which systematically decomposes errors into two main dimensions: representation and measurement. TMLE is represented in diagram of Figure 4.4.

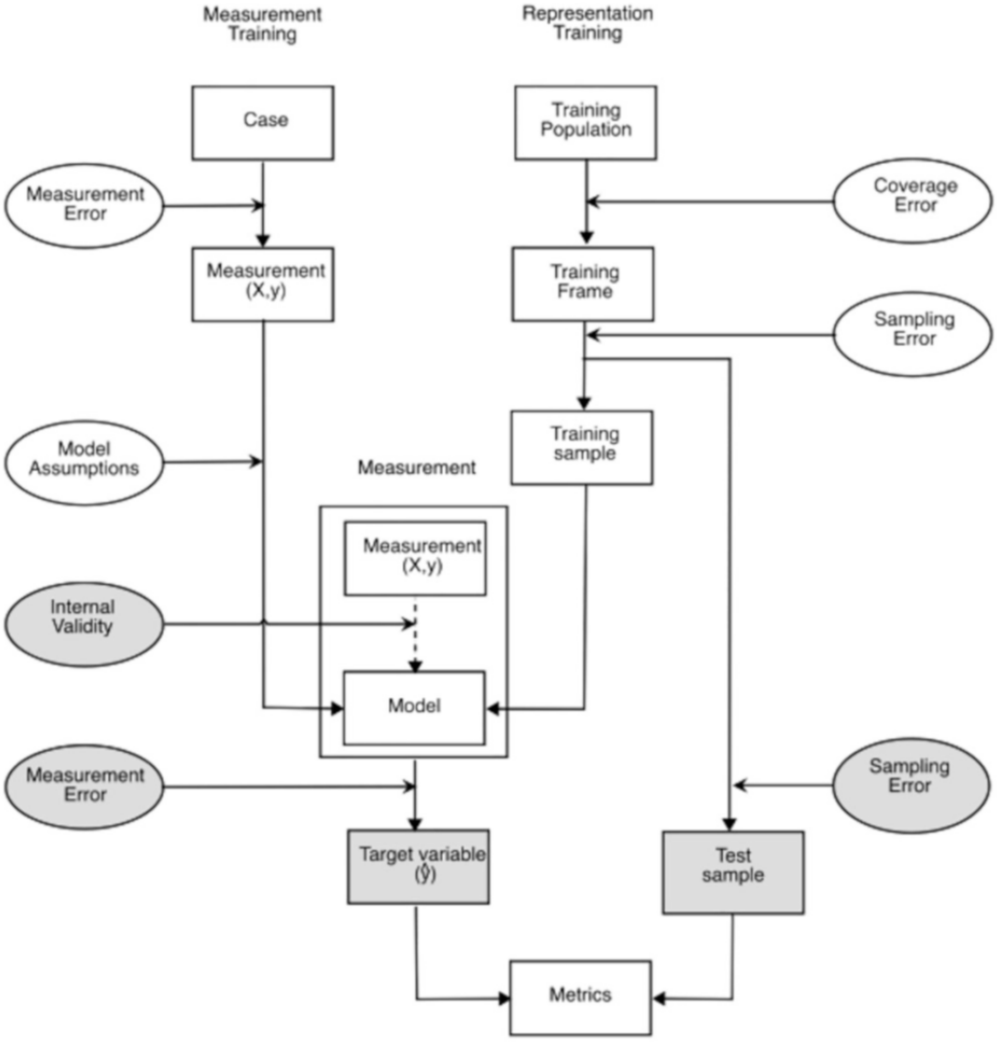

Figure 4.4 The Total Machine Learning Error model: training and testing phases. The training phase is highlighted in white, and the testing phase is highlighted in gray (fig2.3 from Puts et al. [2025])

In the representation dimension, it is analyzed how well the training data and the target data reflect the actual population, accounting for risks such as self-selection bias and coverage gaps in non-traditional data sources. In the measurement dimension, they scrutinize the validity of the input features and, most importantly, the reliability of the "ground truth" or labels used during training, recognizing that errors in the initial classification will inevitably propagate through the model. Furthermore, the authors include a specific "processing and modeling" phase within the TMLE, where they address errors unique to the machine learning lifecycle, such as algorithmic bias, overfitting, and the challenge of concept drift. By synthesizing these elements, the authors provide a robust governance tool that allows National Statistical Institutes to move beyond "black-box" validation, ensuring that ML-based outputs are transparent, reproducible, and meet the high-quality standards necessary for public policy and official reporting.

4.3.1. Micro-level evaluation

Distinguishing Loss Functions and Error Metrics

A critical distinction in model selection is the difference between the functions used to train the model and those used to evaluate its quality. While often related, they serve different purposes in the machine learning pipeline Bishop [2006].

- **Loss Function:** This is the objective function that is minimized during the training phase.
  - It is the mathematical driver of the optimization process; the algorithm adjusts internal parameters specifically to reduce this value.
  - It may not always be intuitively interpretable (e.g., Log-Loss or Cross-Entropy).
  - Different models are inherently tied to specific loss functions (e.g., Support Vector Machines use Hinge Loss, while Linear Regression typically uses Mean Squared Error).
  - The choice of loss function directly impacts how the model learns and its final generalization capabilities Goodfellow et al. [2016].
- **Error Metric:** This is a function used to provide a numerical value that quantifies the quality of the model's predictions for human assessment.
  - It is preferred that metrics are intuitively interpretable by practitioners and stakeholders (e.g., Accuracy or Mean Absolute Error).
  - Metrics are "model-agnostic": they can be used to compare different types of algorithms (e.g., comparing a Random Forest against a Neural Network) regardless of the loss function used during their training.
  - It is standard practice to monitor multiple metrics simultaneously to obtain a multi-faceted view of performance.
  - The choice of metrics depends strictly on the nature of the task, distinguishing clearly between **classification** and **regression** problems Murphy [2012].

Metrics are the quantitative tools used to assess the quality of a model. Without proper metrics, it is impossible to compare different algorithms objectively.

Metrics for Regression:

In regression tasks, we measure the distance between the predicted value ($\hat{y}$) and the actual value (y).

- **Mean Squared Error (MSE):** $\frac{1}{n}\sum(y_i - \hat{y}_i)^2$. It penalizes large errors heavily.
- **Mean Absolute Error (MAE):** $\frac{1}{n}\sum|y_i - \hat{y}_i|$. It is more robust to outliers.
- **R-Squared (R^2):** Represents the proportion of variance for the dependent variable that's explained by the model.

Metrics for Classification:

For classification, performance is often summarized in a **Confusion Matrix**, leading to the following metrics:

- **Accuracy:** The ratio of correctly predicted observations to the total observations.
- **Precision:** The ability of the classifier not to label as positive a sample that is negative.
- **Recall (Sensitivity):** The ability of the classifier to find all the positive samples.
- **F1-Score:** The harmonic mean of Precision and Recall, useful for imbalanced datasets.
- **AUC-ROC:** Measures the ability of a classifier to distinguish between classes across all possible thresholds.
- **Proper Scoring Rules** Gneiting and Raftery [2007]:
  - Brier Score: BS $= \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$, where f_i are the frequencies under the model and o_i the observed values.
  - Logarithmic Score: LS $= -\sum_{i=1}^{N}\log(f_i)$
- **Ranking Metrics:**
  - Area Under Precision-Recall Curve (AUPRC)
  - Lift charts and gains charts

4.3.2. Macro-level evaluation

Let $Y = \{y_1, \ldots, y_N\}$ be true values and $\hat{Y} = \{\hat{y}_1, \ldots, \hat{y}_N\}$ be predictions. Define aggregation function $g : \mathbb{R}^N \to \mathbb{R}^m$. The macro-level error is:

$$E_{\text{macro}} = d(g(Y), g(\hat{Y}))$$

where d is a distance metric appropriate for the aggregated statistics.

Types of aggregates and their evaluation (numeric variables):

- **Linear Aggregates:**

$$S = \sum_{i \in \mathcal{I}} w_i y_i$$

  Relative error: RE $= \frac{|S-\hat{S}|}{|S|}$

- **Absolute Percentage Error (APE)** ([Hyndman and Koehler, 2006]):

$$\text{APE} = \left|\frac{\text{Observed} - \text{Predicted}}{\text{Observed}}\right| \times 100\% \tag{4.4}$$

- **Relative Deviation** (or Bias):

$$\text{Rel. Dev.} = \frac{\text{Predicted} - \text{Observed}}{\text{Observed}} \tag{4.5}$$

- **Nonlinear Aggregates:**
  - Ratios: $R = \frac{\sum_{i \in A} y_i}{\sum_{i \in B} y_i}$
  - Gini coefficient, Theil index, entropy measures
  - Quantiles and interquartile ranges

4.3.3. Model drift

The model drift it is the "expiration date" of a model. Reality is dynamic. If you trained a model to predict house prices in 2019, that model is likely useless in 2024 because the economy (Data Drift) and people's buying habits (Concept Drift) have changed. This forces you to start the Model Selection process all over again.

Model selection is not a one-time event. Even the most accurate model will eventually degrade due to a phenomenon known as **Drift**. Drift represents the change in the relationship between input data and the target variable over time.

In the literature, generally two types are distinguished Gama et al. [2014]:

- **Concept Drift:** The statistical properties of the target variable change (i.e., the "definition" of what we are predicting changes). For example, a fraud detection model may fail if criminals change their behavior.
- **Data Drift (Feature Drift):** The distribution of the input data changes, even if the underlying relationship with the target remains the same. For example, a model trained on data from one demographic may fail if the user base shifts to a different age group.

Detecting drift requires continuous monitoring of the evaluation metrics discussed in the previous section.

4.4. Practical information from the projects

4.4.1. Early imputation in school enrollment (IT)

ML methods are applied to estimate the probabilities of being enrolled in a school course, based on enrollment in the previous school year. These probabilities are then used to predict school enrollment for the subsequent school year. The training dataset, used to estimate the probabilities, consists of individuals enrolled in the school year t-2/t-1 (2020/2021), with the response variable being School Enrollment in the school year t-1/t (2021/2022). In the next step, the estimated probabilities are applied to data shifted by one year: each individual enrolled in a primary or secondary school in t-1/t (2021/2022) is assigned a probability of being still enrolled in t/t+1 (2022/2023).

Since student data are published by Istat at a macro level, the primary goal of this study is to accurately replicate the distribution of the School Enrollment variable at the macro level. However, micro-level accuracy should not be overlooked, as it is essential for evaluating the potential inclusion of this information as a variable in the Base Register of Individuals. To better replicate macro-level accuracy, the value 0 (or 1) of the School Enrollment variable is assigned to each individual by randomly drawing from the estimated probabilities. The experimentation

with ML methods focuses on the application of Random Forests (RF), which also allows for the exploration of feature importance.

In relation to model evaluation, various RF configurations are considered. After preliminary experimentations, the number of trees was fixed at 500. A grid search was performed to identify the optimal values for two parameters: the number of features considered at each split (`mtry`) and the minimum number of samples required in leaf nodes (`min_sample_leaf`). This latter parameter is important to address the problem of imbalance of the response variable.

The performance of these configurations is assessed using confusion matrices computed on the test set. The true values of the response variable are obtained from administrative sources, while the estimated values are derived from a binomial distribution with probability p, where p is obtained from the ML model. When employing Subset 2 of covariates, including all the original variables in the dataset, the RF model always exhibits slightly superior performance, both at the micro level (as indicated by the recall metric) and in terms of distribution.

In this specific case study, the training set is a complete population so the results are not affected by sample variability. The training dataset consists of individuals enrolled in a primary or lower secondary school in the school year t-2/t-1 (2020/2021). The response variable to train the model is the School Enrollment in the school year t-1/t (2021/2022). The test set is composed by individual with the same characteristic of the training set shifted by 1 year: individuals enrolled in a primary or lower secondary school in t-1/t (2021/2022) to which we have to assign a probability of being still enrolled in t/t+1 (2022/2023). This setup closely reflects the real-world scenario, where the model is trained on data from the previous year and then applied to the current year, assuming that the underlying probabilities remain stable from one year to the next. Since the dataset used for the experimentation is built using data from the previous year, the true values for the year of interest are available from administrative sources. Rolling windows cross-validation should be performed to assess the temporal stability of the selected hyperparameters (future work).

To select the final model, the estimated school enrollment is compared to the true values available from administrative sources through the construction of confusion matrices. In particular, three indicators and the computational efficiency are considered:

- **Recall** (also known as sensitivity or true positive rate): It is defined as the ratio between the number of True Positives (TP) identified by the model and the total number of Positives observed:

$$\text{Recall} = \frac{\#\text{True Positives}}{\#\text{True Positives} + \#\text{False Negatives}} \tag{4.6}$$

  It measures how well a model identifies positive cases. In simple terms, recall tells us the proportion of actual positives that the model correctly identified. A high recall means the model misses very few positive cases.

- **Precision**: It is defined as the ratio between the number of True Positives (TP) identified by the model and the total number of Positives identified by the model ($TP + FP$):

$$\text{Precision} = \frac{\#\text{True Positives}}{\#\text{True Positives} + \#\text{False Positives}} \tag{4.7}$$

  It measures the accuracy of the positive predictions made by a model. In simple terms, precision tells us how many of the instances predicted as positive were actually positive. A high precision means that when the model predicts a positive case, it's usually correct.

- **Difference between frequencies**: It is defined as the difference between the percentage of predicted Positives and the percentage of observed Positives in the reference population:

$$\text{Difference} = \frac{\#\text{Predicted Positives} - \#\text{Observed Positives}}{\#\text{Reference population}} \tag{4.8}$$

  It measures the aggregate-level accuracy (or bias) of the predictions. In simple terms, it tells us how far off the model is in estimating the total number of positive cases. A small difference indicates that the model accurately predicts the overall number of positive cases, even if individual classifications may be incorrect. Since official statistical outputs are mostly required at the aggregate level, this is a particularly relevant indicator.

- **Relative difference between frequencies**: It is defined as the difference between the percentage of predicted Positives and the percentage of observed Positives, normalized by the percentage of observed Positives:

$$\text{Relative Difference} = \frac{\#\text{Predicted Positives} - \#\text{Observed Positives}}{\#\text{Observed Positives}} \tag{4.9}$$

  It measures the aggregate-level accuracy (or bias) of the predictions accounting for the scale of the quantities being compared. This makes comparisons more meaningful when units differ or when variables have very different sizes.

- **Run-time**: Beyond predictive performance, the computational efficiency and the time required for the model to complete a training and inference cycle are also considered in the choice of the final model.

In Random Forest, the best parameter settings for dataset 1 is with `mtry` =8 and `min_sample_leaf`=1. With this configuration Recall=26,5%, Precision=25,9%, Difference=0,028% and the running time to train the model is 19 minutes. The choosen parameter settings for dataset 2 is with `mtry` =20 and `min_sample_leaf`=1. With this configuration Recall=56.0%, Precision=55,9%, Difference=0,002% and the running time to train the model is 24 minutes.

In relation with the drift of the model, every year the selected model should be tested on the last available administrative data in order to assess the stability of the phenomenon and the relationship between variables.

4.4.2. Early imputation in the industrial turnover index in Germany (DE)

The evaluation process considered several machine learning algorithms, including **Random Forest, Linear Regression, CART, and XGBoost**. Among these, four specific modeling procedures were analyzed in greater detail, categorized into one-step and two-step imputation workflows:

1. **One-step procedure:**

   a) Random Forest: The target variable is imputed in an iterative procedure together with the regressors using a Random Forest model Breiman [2001]. This was implemented via the R package `missRanger` [Mayer, 2023].

   b) Linear Regression: The target variable and regressors are imputed simultaneously in an iterative procedure through a linear regression model. This was executed using the R package `mice` (Multivariate Imputation by Chained Equations) [van Buuren and Groothuis-Oudshoorn, 2011], see also for more detail van Buuren [2023]. For the experiments, mice was used as a single imputation method (with parameter $m = 1$), although the use of multiple imputations for uncertainty quantification could be explored in future work.

2. **Two-step procedure:**

   a) **Random Forest with imputed regressors:** This approach involves an initial imputation of the regressors using the R package `mice` via linear regression, followed by the application of the `ranger` package [Wright and Ziegler, 2017] to estimate the target variable using the Random Forest algorithm.

   b) **Linear regression with imputed regressors:** This consists of imputing the regressors using linear regression through the `mice` package, and subsequently applying a linear regression model from the `caret` package [Kuhn, 2024] to estimate the final target variable.

Regarding the target variable definition, estimating domestic and foreign turnover individually and subsequently aggregating them did not yield improvements in the accuracy metrics. Consequently, the models focused on the total turnover estimation directly.

The cross-validation method has been rolling backtesting by comparing monthly estimates with later received ground truth (to create a realistic testing scenario).

The final model selection is based on the prediction of the total turnover at an aggregate level, encompassing all companies and economic sectors. The target variable is defined as the sum of **EF18** (domestic turnover) and **EF19** (foreign turnover).

To assess the accuracy and bias of the models, two primary evaluation metrics are utilized: **Absolute Percentage Error (APE)** and **Relative Deviation**. These metrics are computed across two distinct temporal windows to evaluate the impact of data availability on model quality:

- **Window 1 (July 2022 – July 2023):** This scenario uses data available at $t + 15$ days. The objective is to determine if an early estimation, conducted only 15 days after the reference period, yields sufficient quality for official statistics.
- **Window 2 (July 2022 – February 2024):** This scenario uses data available at $t + 20$ days. This window benefit from a larger volume of observed data and a lower proportion of estimated values, typically resulting in higher stability.

The final selection metric for model accuracy is determined by calculating the **mean of the error metrics** over all months within the specified windows. This provides a single, robust value that represents the consistent performance of the candidate model over time.

The model finally selected is linear regression (mice – norm.predict).

The economic sectors differ in terms of turnover, company structure, volatility, and their percentage of the manufacturing industry. Due to this heterogeneity, the imputation was carried out on an industry-specific basis at the 2-digit level to adequately take industry-specific patterns into account. (Not separating individual models by economic sector, but instead estimating separately by company size class, delivered significantly worse results.)

4.4.3. Early imputation in the industrial turnover index in Spain (ES)

In the Spanish case, we have prioritized building an end-to-end prototype production process that includes every aspect required for future implementation under real-world conditions, leaving the optimization and selection of the model for later stages. Nevertheless, we have taken several factors into account.

Firstly, we pursue versatility so that any type of regressor can be used in the model. Furthermore, non-linear dependence between the target variable and the regressors must be allowed. Thus, our first choice has been to use Random Forests [see e.g. Hastie et al., 2009, Murphy, 2012].

Secondly, in order to improve accuracy and, especially, to deal with outliers (that are very important in the index as we have already explained) which will be rapidly detected through their residuals, we have chosen to use boosting [see e.g. Watt et al., 2020]. In this way, the trained boosted regression tree will naturally incorporate the effect of outliers, which will also be predicted with a reasonable accuracy.

Finally, among the different choices within the boosting algorithm family we focus on the gradient boosting algorithm [Friedman, 2001] and, in particular, on the LightGBM version [Ke et al., 2017] [see also Microsoft Corporation, 2022]. This choice is basically motivated by speed without a compromise of accuracy [see Bentéjac et al., 2021, for a comparison of gradient boosting algorithms]. We have used the R API in the form of the R package `lightgbm` [Shi et al., 2021]. Over the last few years, this algorithm and its package have evolved, leading to some discrepancies in the results. It is important to be cautious regarding changes across different versions of the implementation. Parameters and hyperparameters are specified below.

In our case, the calendar time and the monthly periodicity plays a central role in data availability and data use. To predict turnover values at time d in reference month m and year y, we must be aware that validated values from the preceding reference month $m-1$ are already available at these time instants. Thus, we follow for each and every month m the following temporal cross-validation procedure:

- We train the model for a set of multiple alternative hyperparameter sets $h = 1, 2, \ldots$ with data up to reference month $m-2$.
- We apply each trained model to the data set with corresponding reference month $m-1$ obtaining, thus, the predicted values $\widehat{z}_k^{m-1y}(d)$ for each unit k.
- We compute for each trained model ξ_h the absolute error of the total turnover $AE_h = \left| \sum_{k \in U^{my}} \widehat{z}_k^{m-1y,\xi_h}(d) - \sum_{k \in U^{my}} z_k^{m-1y,\text{val}}(d) \right|$. We select the model m_{h^*} with optimal value of AE_h.
- We train again the same model with data up to reference month $m-1$ and the collected units of reference month m at time d with hyperparameters h^*.
- We apply the trained model to data not yet collected up to time d of reference month m and year y. Thus, we obtain the predicted values $\widehat{y}_k^{my,\xi_{h^*}}(d)$ to be plugged in the estimator.

One might note that our approach omits a traditional test set in favor of training and validation subsets. This is justified by two factors: first, the pilot study simulates the exact operational workflow used in production, where the data at reference period serves as the de facto test set for direct comparison with ground-truth indices. Furthermore, the recursive nature of our monthly updates, incorporating fresh data over a multi-month horizon, provides strong evidence that the model generalizes well without overfitting.

Consistent with the statistical learning approach, a specific set of hyperparameters must be tuned prior to training the predictive model. For this pilot study, an exhaustive search for optimal values was not conducted; instead, priority was given to establishing a viable, end-to-end production pipeline. This framework is designed to be iteratively refined and adaptable to evolving operational conditions. Consequently, we have employed a minimal hyperparameter grid focusing on two primary parameters (see Table 4.1), namely `nrounds` (the number of training rounds) and `eta` (the shrinkage rate in the gradient boosting algorithm).

nrounds	eta
300	0.05
1000	0.05
300	0.01
1000	0.01

Table 4.1 Minimal hyperparameter grid search

Three more hyperparameters are customised in the core function `lightgbm` according to the following values to train the models:

Parameter	Definition	Value
'objective'	Type of regression application	'regression'
'metric'	Metric to be applied on the evaluation set(s)	'mae' (absolute loss)
'boosting'	Algorithm variant	'gbdt' (traditional Gradient Boosting Decision Tree)

Table 4.2 Customised hyperparameters for function `lightgbm`

The rest of parameters takes on their default values [see Microsoft Corporation, 2022, for details]. Missing values treatment and encoding are not part of hyperparameter optimization. Needless to say, our choices are clearly suboptimal and a more exhaustive search should be accomplished to find the best combination of hyperparameters, missing values treatment and encoding, especially those regarding a trade-off between accuracy and speed [see Parameter Tuning section in Microsoft Corporation, 2022]. Despite this, results prove that an evolving end-to-end process can be effectively designed and implemented to provide reasonably accurate early estimates.

In relation to the quality evaluation, the performance of the prediction model is assesed by direct comparison between each predicted set of indices for the complete range of breakdowns published in the press release with their truly released versions under the traditional production process with the whole data collection and data editing phases fully accomplished.

Thus, if we denote by $\widehat{I}_{U_A}^{my}(d)$ the predicted ITI for domain U_A and reference period my at day d, we compute $\widehat{I}_{U_A}^{my}(d) - I_{U_A}^{my}$. We proceed similarly with the monthly and annual rates:

$$\widehat{\Delta}_{m,U_A}^{my}(d) = \frac{\widehat{I}_{U_A}^{my}(d) - \widehat{I}_{U_A}^{m-1y}(d)}{\widehat{I}_{U_A}^{m-1y}(d)}, \qquad \widehat{\Delta}_{y,U_A}^{my}(d) = \frac{\widehat{I}_{U_A}^{my}(d) - \widehat{I}_{U_A}^{my-1}(d)}{\widehat{I}_{U_A}^{my-1}(d)}.$$

Besides this empirical approach, it is highly convenient to make a theoretical analysis of accuracy to understand those factors impinging on it. We detach this analysis on the properties of bias and mean squared error of the proposed estimator.

4.4.4. Early imputation in accommodation establishments (PL)

4.4.4.1. Introduction and Applied Models

In this study, various machine learning models with different levels of complexity were tested, ranging from stepwise regression and the LASSO model, through Decision Trees, to Random Forest. Stepwise regression and the LASSO model include a variable selection step

and, in practice, overfitting rarely occurs. Therefore, no additional validation procedures, such as cross-validation, were applied to assess these two models.

For the Decision Tree, k-fold cross-validation was applied using the built-in mechanism of the R package `rpart`. This procedure allowed identification of the optimal complexity parameter, followed by tree pruning. Similarly, in the Random Forest model, the decision criterion was the Out-Of-Bag (OOB) error, which is also an inbuilt procedure in the R package `randomForest`.

The performance of the tuned models was evaluated using several metrics relevant for numerical data, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and MAPE, as well as considering the similarity of distributions.

4.4.4.2. LASSO Model

The regression model was fitted using the LASSO (Least Absolute Shrinkage and Selection Operator) method, which allows both coefficient regularization and automatic selection of important variables.

First, the design matrix `x` was prepared using the `model.matrix` function for all independent variables in the `data_clean2` dataset, excluding the intercept column, along with the dependent variable vector `y` corresponding to the number of tourists. Cross-validation was then applied using the `cv.glmnet` function with `alpha = 1`, corresponding to the L1 penalty typical for LASSO. Based on the value of `lambda.min`, which minimizes the validation error, the model coefficients were extracted, and variables with non-zero coefficients (excluding the intercept) were selected.

Next, a linear regression formula was dynamically created in R by combining the selected variables, and the linear model `lm_model_lasso` was fitted. The analysis showed that the model fits the data well, as indicated by the high coefficient of determination $R^2 = 0.8885$. Among the selected variables were `tourists_minus1`, `tourists_minus2`, various winter average values by region and facility type, and the 12-month tourist share relative to the province.

To assess forecast quality, basic error metrics were calculated: RMSE = 268.1, MAE = 117.7, MAPE could not be computed due to the presence of zero values in the dependent variable, and the mean error (ME) was practically zero. These values were saved to an Excel sheet to enable comparison with other models.

The application of LASSO thus allowed not only limiting overfitting through coefficient regularization but also selecting the most relevant variables explaining the number of tourists, facilitating model interpretability and its practical use in tourism demand forecasting.

Metric	Value
RMSE	268.092445
MAE	117.743074
MAPE (%)	–
ME	-1.3949×10^{-12}

Table 4.3 Basic forecast statistics of the LASSO model

Statistic	Value
Residual standard error	268.2 on 13,914 DF
Multiple R-squared	0.8886
Adjusted R-squared	0.8885
F-statistic	2.219×10^4 on 5 and 13,914 DF
p-value	$< 2.2 \times 10^{-16}$

Table 4.4 LASSO regression model fit statistics

4.4.4.3. Decision Tree Model

To analyze the relationship between the number of tourists and the set of explanatory variables, a regression decision tree was applied. First, the dataset `data_clean3` was prepared, excluding any missing values. Then, the decision tree model was fitted using the `rpart` function, setting the minimum complexity parameter `cp = 0.001`, which allowed the tree to grow relatively deep to capture significant patterns in the data.

To select the optimal pruning level, the `cptable` table and the cross-validation error plot (`plotcp`) were used. Determining the minimum prediction error enabled identification of the optimal `cp` value, according to which the tree was pruned to a more stable and less overfitted form. The final tree was visualized using the `rpart.plot` function, which facilitated easy identification of the most important variables in the model.

Variable importance analysis highlighted which features had the greatest impact on the predicted number of tourists, which is particularly useful for model interpretation. After fitting the pruned tree, predictions were made for the entire dataset, and model accuracy was evaluated using error metrics: RMSE = 257.1, MAE = 117.8, and MASE = 0.227, indicating good forecasting performance relative to a simple naive model. MAPE and MPE were high due to the sensitivity of these metrics to small values in the dependent variable.

The use of decision trees allows not only modeling of nonlinear relationships between variables but also provides an intuitive interpretation – it is easy to see which variables and their values determine the predicted number of tourists. Pruning the tree according to the optimal `cp` parameter reduced overfitting while retaining key relationships in the data. The results were saved to an Excel sheet for comparison with other models, enabling evaluation of the performance of different forecasting methods.

4.4.4.4. Support Vector Machine (SVM) Model

In the next step of the analysis, a Support Vector Machine (SVM) was applied for regression of the number of tourists. First, a linear SVM model was fitted using the `svm` function from the `e1071` package with the `eps-regression` type. Initial predictions were made for all observations, followed by parameter tuning through cross-validation for different values of the `cost` parameter (10 and 100) to identify the model that best fits the data. The best linear model obtained during tuning was then used to predict the number of tourists for the entire dataset. Forecast evaluation indicated that the linear model achieved RMSE = 270.3, MAE = 107.2, MAPE = 212.8%, ME = 2.08, and MSE = 73,074, showing moderate predictive performance of the linear SVM.

Subsequently, an SVM with a radial basis function (RBF) kernel was applied, which allows modeling of nonlinear relationships between variables. Parameter tuning in this case included both `cost` (10 and 100) and `gamma` (0.01 and 0.1), enabling adjustment of model flexibility to the data characteristics. The radial SVM model showed substantially better performance compared to the linear variant – RMSE decreased to 208.3, MAE to 74.5, MAPE to 172%, ME = 6.38, and

MSE = 43,403. These results indicate that accounting for nonlinearity in the model significantly improved forecast accuracy and better captured the variability in the number of tourists.

The use of support vector machines, both linear and nonlinear, allowed capturing different types of relationships in the data. Linear dependencies are well modeled by the linear SVM, while the radial SVM is able to detect complex interactions and nonlinear patterns, as evidenced by the substantial improvement in error metrics compared to the linear model. All forecast statistics were saved to an Excel sheet for later comparison with the results of other models.

4.4.4.5. Multivariate Adaptive Regression Splines (MARS) Model

Multivariate Adaptive Regression Splines (MARS) is a nonlinear regression method that fits the model to the data by combining simple basis functions (splines). This method is particularly useful when relationships between independent variables and the dependent variable are nonlinear or when there are significant interactions among predictors. MARS automatically identifies knots and interactions between variables and allows assessment of predictor importance using the `evimp` function. This makes the model both flexible and relatively easy to interpret.

In the analysis of the number of tourists, the MARS model was fitted using the `earth` package in R, and variable importance was assessed with the `evimp` function, allowing identification of the most relevant predictors. Based on these predictors, a simplified linear model was constructed using only the most important variables recommended by MARS. Plots of the first basis functions and interactive visualizations of all predictors further facilitated the interpretation of each variable's influence on the predicted number of tourists.

Predictions obtained with the MARS model showed high accuracy. The evaluation metrics are as follows: RMSE = 254.345, MAE = 109.470, ME practically zero (-1.714×10^{-11}), MAPE = 175.534%, and MSE = 64,691.342. For the linear model fitted only with variables suggested by MARS, the metrics were slightly worse: RMSE = 269.543, MAE = 119.034, ME practically zero (-1.157×10^{-12}), MAPE = 281.335%, and MSE = 72,653.322. These differences indicate that MARS better handles nonlinearity and detects important interactions between predictors, which a simple linear model cannot reflect.

Minimal ME values in both models suggest no systematic bias in the predictions, whereas higher MAPE values in the linear model indicate worse fit for variables with high variability. MARS also allows identification of the most important variables affecting the number of tourists, which is useful for both data interpretation and constructing simpler regression models.

In summary, the MARS model proved to be an effective tool for analyzing tourism data, improving forecast accuracy compared to classical linear regression. Its ability to automatically detect nonlinear relationships and interactions, as well as assess variable importance, combines the flexibility of nonlinear models with interpretability, making MARS a valuable tool in quantitative and predictive research.

Metric	Value
RMSE	254.3449
MAE	109.4701
ME	-1.714×10^{-11}
MAPE (%)	175.5338
MSE	64,691.3424

Table 4.5 Prediction quality metrics for the MARS model

Metric	Value
RMSE	269.5428
MAE	119.0336
ME	-1.157×10^{-12}
MAPE (%)	281.3349
MSE	72,653.3220

Table 4.6 Prediction quality metrics for the linear model based on variables selected by MARS

4.4.4.6. Boruta Feature Selection

To identify the most important predictive variables for the number of tourists, the Boruta algorithm was applied. Boruta is an all-relevant feature selection wrapper method. It uses Random Forest models to assess the importance of each variable and classifies them as Confirmed (important), Rejected (not important), or Tentative (uncertain).

Before analysis, observations with zero tourists were excluded, creating the subset `data_clean4`. Then, the Boruta model was fitted using the `Boruta()` function from the `Boruta` package, setting `maxRuns = 150` to ensure stable results and `doTrace = 2` to monitor the algorithm's progress. Execution time was measured to assess computational efficiency.

Boruta results indicated a set of variables confirmed as important for predicting the number of tourists. Tentative variables were subjected to the `TentativeRoughFix` procedure, ultimately eliminating irrelevant features and retaining only those with real contribution to the model. For visualization, bar plots were created showing the mean importance (`meanImp`) and Boruta decisions, both for all predictors and limited to the 15 most important features. Bar colors indicated variable status: Confirmed – green, Rejected – red, Tentative – gold.

Using the variables confirmed by Boruta, a Random Forest model with 500 trees was built. Predictions for the number of tourists were made for the entire `data_clean4` dataset. Forecast quality was then assessed using standard metrics: RMSE, MAE, ME, MAPE, and MSE.

The obtained results for the Random Forest model on Boruta-selected variables are: RMSE = 99.502, MAE = 35.444, ME = -1.393, MAPE = 14.940%, and MSE = 9,900.678. The high predictive accuracy indicates that Boruta effectively selected relevant variables and that the Random Forest efficiently utilized them to predict tourist numbers.

In practice, Boruta proved to be an effective feature selection method, eliminating irrelevant and uncertain variables, which enabled building a more stable and interpretable predictive model. Combining Boruta with Random Forest provided both high prediction accuracy and clear interpretability of the influence of individual variables on tourist numbers, making this approach a valuable tool for tourism data analysis and other complex datasets.

Metric	Value
RMSE	99.5021
MAE	35.4443
ME	-1.3935
MAPE (%)	14.9401
MSE	9,900.6778

Table 4.7 Prediction quality metrics for the Random Forest model on Boruta-selected variables

5

Results

In this chapter we will see some results from each project. Here, a brief summary of the results is shown:

Italy (IT-SchoolEnrollment): The experimentation proved successful, demonstrating that machine learning can outperform standard logistic regression. The **Random Forest** model emerged as a suitable approach in the presence of imbalanced data, with the rare event supported by a non-negligible number of observations. Hyperparameter tuning was aimed at balancing micro- and macro-level performance, with optimal configurations depending on the degree of class imbalance and the informativeness of the covariates. When the discriminatory power of the regressors is limited, deeper trees improve performance by better exploiting weak signals. Conversely, when information is concentrated in a few predictors, the tuning of mtry becomes more critical than tree depth.

Germany (DE-ITI): The project is considered a success and has been published as experimental data since January 2025. Interestingly, the **one-step linear regressio**n (using the mice package) was selected as the optimal method over non-linear machine learning models. This choice was driven by the strong linear relationship inherent in the data and the method's superior robustness, simplicity, and maintainability. A critical observation was that estimates at $m + 20d$ significantly outperform those at $m + 15d$ due to the higher availability of edited data.

Spain (ES-ITI): This project successfully established a pipeline for daily early estimates using **Gradient Boosted Decision Trees** (GBDT). The model proved robust, with hyperparameters remaining stable over time, suggesting that cross-validation frequency could be reduced. A major methodological insight was that retraining the model with newly collected units from the reference period provides a much greater boost to accuracy than the inclusion of current-period regressors, highlighting the vital importance of real-time data flow.

Poland (PL-Accommodation): The results indicate that the **Random Forest** model combined with Boruta feature selection is highly effective at capturing the strong seasonal and regional patterns of the tourism sector. The project is considered successful for regular forecasting; however, a notable finding was the model's struggle with the disruptive event of the COVID-19 pandemic. While it accurately tracks stable seasonal peaks, it tends to overestimate during sudden, extreme drops in activity, indicating a need for specific adjustments during anomalous periods.

5.1. Early imputation in school enrollment (IT)

In this case study, the primary objective was to improve the predictive performance of the baseline model, specifically a logistic regression adjusted for rare events, while also employing Random Forests as a tool for methodological insight and exploratory data analysis.

Initial experiments applying the Random Forest model on Dataset 1, with the same set of covariates used in the logistic regression produced results broadly comparable to those obtained with the standard method. However, improved performance was observed when allowing for deeper trees (Table 5.1). Further gains were achieved when the model was estimated using the original covariates, rather than versions recoded into a smaller number of categories for inclusion in the logistic regression.

Table 5.1 Micro-level accuracy (recall and precision) and macro-level accuracy (difference between % and relative difference between %) for Dataset 1 obtained using logistic regression and Random Forest. Random Forest models were estimated with two values for the minimum number of observations in terminal nodes (MinLeaf 50 and 1) and two sets of predictors: the subset used in the logistic regression (Subset 1) and the full set of original variables (Subset 2).

	Logistic	Random Forest			
		Subset 1 (as in Logistic)		Subset 2 (original variables)	
		MinLeaf = 50	MinLeaf = 1	MinLeaf = 50	MinLeaf = 1
Recall	0.258	0.237	0.255	0.240	0.263
Precision	0.242	0.224	0.244	0.233	0.257
Diff. between %	0.079	0.068	0.056	0.037	0.029
Rel. diff. between %	6.404	5.535	4.553	3.029	2.320

The purpose of this preliminary analysis was to assess whether machine learning methods could potentially achieve higher accuracy than the standard approach, thereby motivating further investigation into their use and properties. At this stage, only a limited exploration of hyperparameters was conducted. Specifically, the number of trees was fixed at 500 and the number of covariates considered at each split was set to 7, approximately equal to the square root of the total number of covariates. Two alternative values were considered for the minimum number of observations (*MinLeaf* in the table) in the terminal nodes (50 and 1), corresponding to different levels of tree depth.

Having established the potential of machine learning methods, a more systematic search for optimal hyperparameters was subsequently carried out in order to further improve model performance, considering both micro- and macro-level evaluation metrics. The variables available in the two datasets (Dataset 1: primary and lower secondary school; Dataset 2: upper secondary school), together with the covariates used in the experimented models, are reported in Appendix A.

With the number of trees held constant (500 trees), the performance of the Random Forest was evaluated by varying the number of instances per terminal node (`min_sample_leaf`) and the number of features considered at each split (`mtry`). Systematic examination of these key tuning parameters highlighted important characteristics of the data structure and provided a clearer understanding of how information is distributed across the predictors.

As shown in Figure 5.1 (left), for dataset 1, which contains a rare class, precision and recall are highest when the minimum leaf size is set to 1, making the effect of `mtry` negligible. For larger leaf sizes (8, 10, 15, 20), performance differences across values of the maximum number of features become more evident, with a clear advantage when this parameter is set to 20. This

suggests that increasing the tree width at each split compensates for the reduced depth, allowing a broader exploration of the predictors.

For dataset 2, characterized by class imbalance, Figure 5.1 (right) shows that, irrespective of the minimum leaf size, higher values of `mtry` lead to the best performance. In this setting, where more informative predictors are available, the model benefits from a greater probability of selecting them—something that occurs when the number of candidate features is large. Consequently, very deep trees may not be required to achieve strong predictive accuracy. The analysis revealed that, in dataset 1, informative patterns were weakly distributed across many features. This insight guided our decision to use deeper trees, enabling the model to fully exploit the dispersed information across regressors and thereby improve predictive performance. For dataset 2, the analyses suggest that a high value of `mtry` should be prioritized, while the minimum leaf size appears less critical. The final parameter selection is guided by a third performance measure: the relative difference between distributions, which serves as an indicator of model quality at the macro level.

Figure 5.1 Micro level accuracy, recall and precision, for dataset 1 (left) and dataset 2 (right) varing `min_sample_leaf` and `mtry`

The analysis of the model's ability to reproduce the marginal distributions of the response variable highlights differences in macro-level performance between the two datasets (Figure 5.2). For dataset 1 , with a minimum leaf size of 1, increasing the number of features per split leads to greater divergence between distributions (higer relative difference between percentage, see left side of Figure 5.2) thereby reducing macro accuracy. This is likely due to the relatively low informativeness of the available predictors; considering a larger number of features introduces noise into the estimates and increases the risk of overfitting.

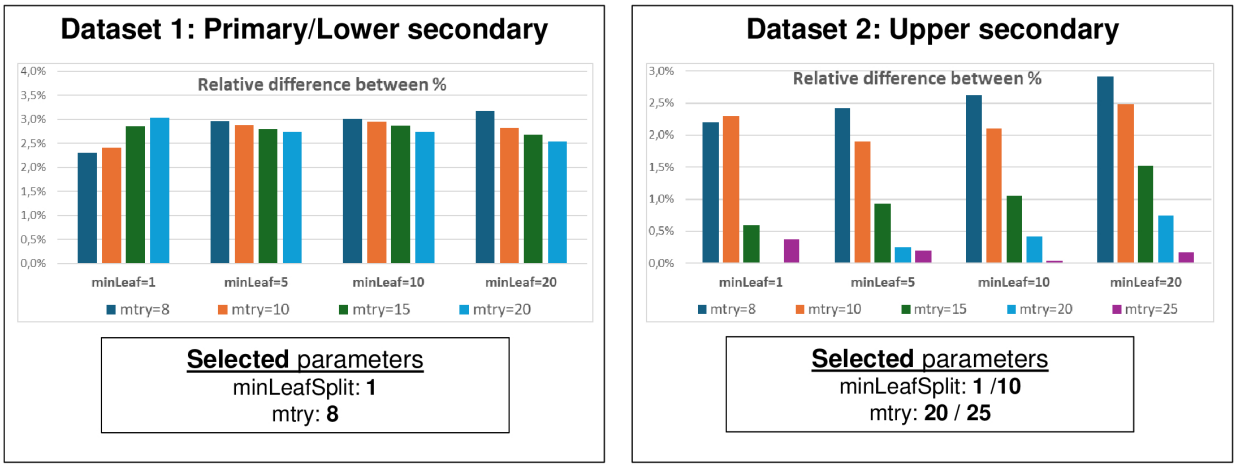

Figure 5.2 Macro level accuracy, relative difference between percentage, for dataset 1 (left) and dataset 2 (right) varing `min_sample_leaf` and `mtry`. Selected parameters.

In contrast, for dataset 2, macro performance improves markedly as `mtry` increases. In this case, several predictors are informative with respect to the outcome, making it important to consider them at each split. Overall, the analysis across the three performance indicators—at both micro and macro levels—suggests different optimal parameter settings for the two datasets: `min_sample_leaf` = 1 and `mtry` = 8 for dataset 1, and `min_sample_leaf` = 1 and `mtry` = 20 for dataset 2. An alternative configuration for dataset 2 (`min_sample_leaf` = 10, `mtry` = 25) yields roughly equivalent performance.

5.2. Early imputation in the industrial turnover index in Germany (DE)

The performance of different imputation methods is evaluated by analyzing both the absolute percentage deviation and the percentage deviation for estimates at time $t+15$ and $t+20$ (see Figure 5.3). Comparing these two points in time shows that estimates at $t+20$ are consistently more accurate across all methods. This improvement is expected, as the share of edited data available at $t+20$ is on average about 20 percentage points higher, reducing the proportion of values that need to be imputed and providing more up-to-date information for model training. As a result, estimates produced five days later are substantially more precise, with improvements ranging between approximately 0.7 and 1.1 percentage points depending on the method. This finding supports the choice of producing estimates at time $t+20$.

With regard to absolute percentage deviation, linear models outperform non-linear approaches. This can largely be attributed to the strong linear relationship between the explanatory variables and the target variable. In addition, non-linear methods such as random forests are restricted to predicting values within the observed range of the training data, whereas linear regression models allow for extrapolation beyond this range. Among the methods considered, the one-step linear regression approach and the two-step approach with linear regression and imputed regressors show the best performance, also reflected in lower variability and smaller maximum deviations, indicating greater robustness.

The analysis of percentage deviation suggests that, for estimates at time $t+20$, all methods are approximately unbiased, as the average deviation is close to zero. Hence, no systematic tendency toward overestimation or underestimation is observed.

The final selection of the imputation method takes into account both empirical performance and additional quality criteria such as reproducibility, timeliness, and interpretability. While both leading approaches perform similarly in terms of reproducibility and timeliness, the one-step linear regression method offers advantages in terms of transparency, ease of implementation, and maintenance. Since all features are imputed jointly within a single model, the procedure is more straightforward and less complex compared to the two-step approach, which requires additional preprocessing steps. Furthermore, the one-step method has been found to be more robust in situations with higher levels of missing data. For these reasons, the one-step linear regression approach is selected as the preferred method.

After evaluating various imputation methods on historical test data and selecting the final approach, the chosen method is subsequently applied to more recent data (see also Yadegar et al., 2025).

Figure 5.4 compares the month-on-month change rates of the $t+20$ estimates with the provisional official results published at $t+45$ over the full observation period. This comparison provides insights into the accuracy of early estimates and highlights deviations between early and later results. Larger discrepancies tend to occur in months with unusually low reporting rates or when data from individual German federal states are missing. The figure also illustrates the substantial month-to-month volatility in turnover, which represents a key challenge for the estimation process.

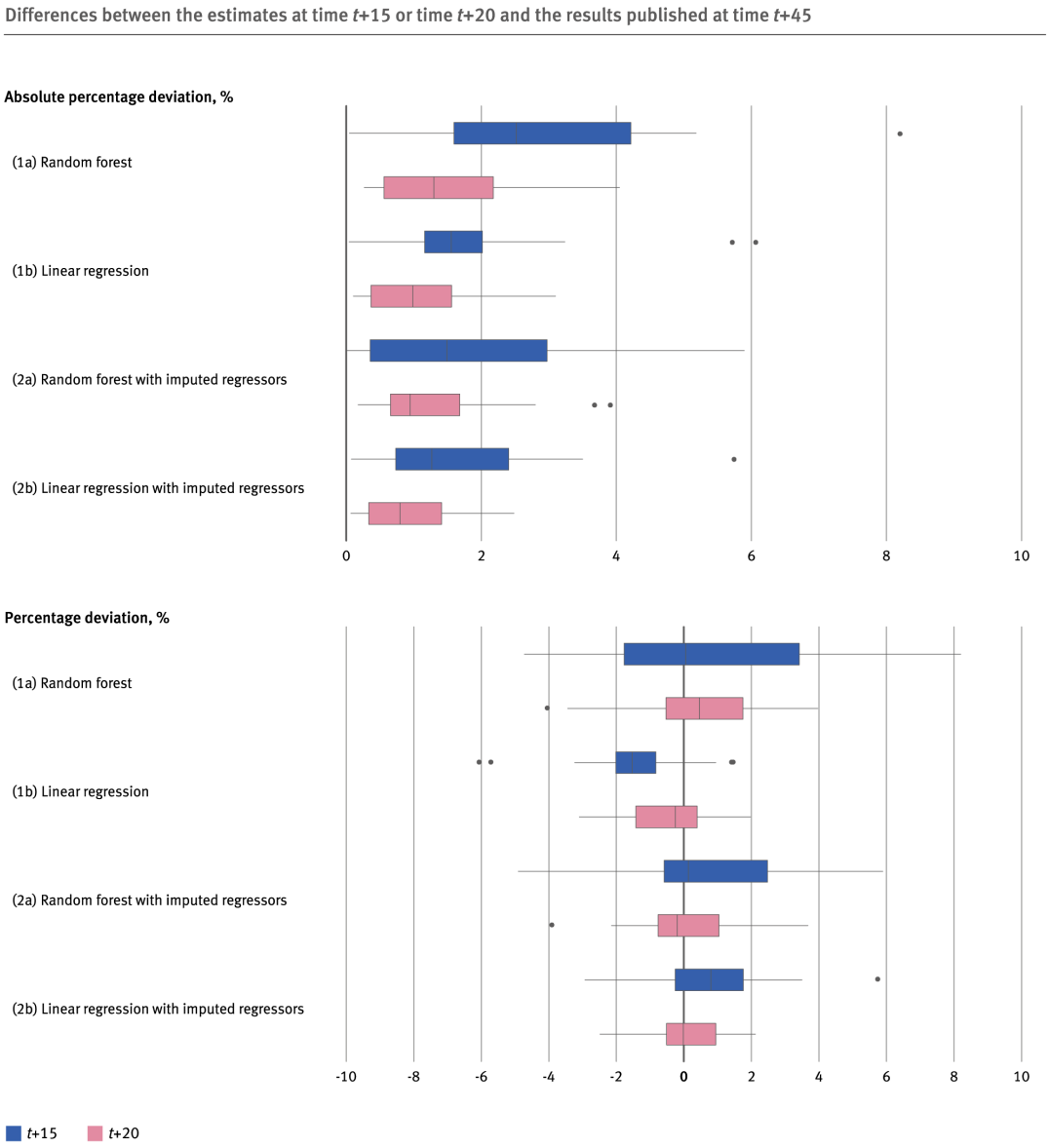

Figure 5.3 Differences between the early estimates and the final published

The $t + 20$ project demonstrates that it is feasible to produce reliable early estimates of short-term economic indicators using microdata-based models. In this context, both regression-based imputation procedures and machine learning methods are examined. The quality of these estimates depends strongly on the availability of already edited reports. While estimates at $t+20$ show clear improvements compared to earlier estimates at $t+15$, challenges remain, particularly in the presence of economic shocks, larger data gaps, and highly volatile or heterogeneous local units.

Time-dependent regressors prove useful under normal economic conditions, as they capture cyclical developments effectively. However, their performance is limited in the case of sudden economic disruptions. Although they can provide early signals of emerging crises, reliable estimates typically only become possible after some delay. In addition, estimation quality varies across economic activities, suggesting that sector-specific modelling approaches—such as the use of activity-specific regressors—could further improve results.

Overall, the project provides robust results for the accelerated production of short-term economic indicators. At the same time, continued methodological development remains important to further enhance estimation quality. In particular, ensuring reliable estimates during periods of economic instability and improving the timeliness and relevance of statistical outputs are key objectives. The project has been published as experimental data since the reference month January 2025 in the EXSTAT section of Destatis. Further information on methodology and background is also expected to be made available there. In addition, estimates of absolute turnover can be accessed via the Dashboard Deutschland, specifically in the "Dashboard Konjunktur" section (see also Yadegar et al., 2025).

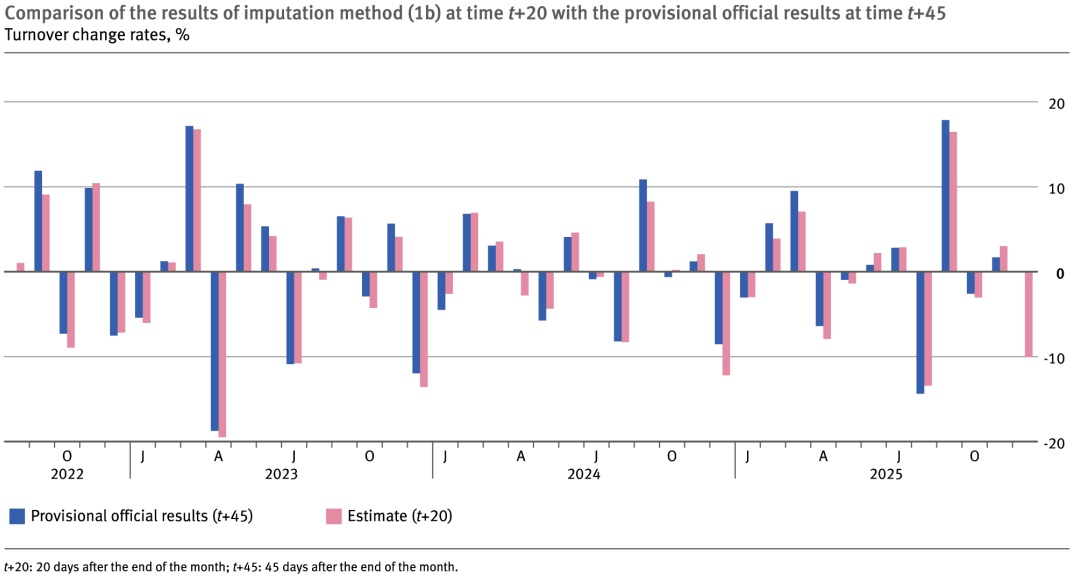

Figure 5.4 Results of imputation done with linear regression at time t+20 (pink) in comparison with t+45 (blue)

5.3. Early imputation in the industrial turnover index in Spain (ES)

The outputs of this project consist of a series of early ITI estimates, disaggregated by standard production conditions, alongside their corresponding monthly and annual variation rates for the three batches overseen by survey management. These metrics are presented in conjunction with their respective Root Mean Squared Errors (RMSE). To further evaluate the model's capabilities, we also provide comparative series including predictions that omit current-period regressors, results from training sets excluding newly collected units, and the definitive index values released at $m + 51d$. The main results are shown in the following and more detailed information about this project can be found in Barragán et al. [2022][1].

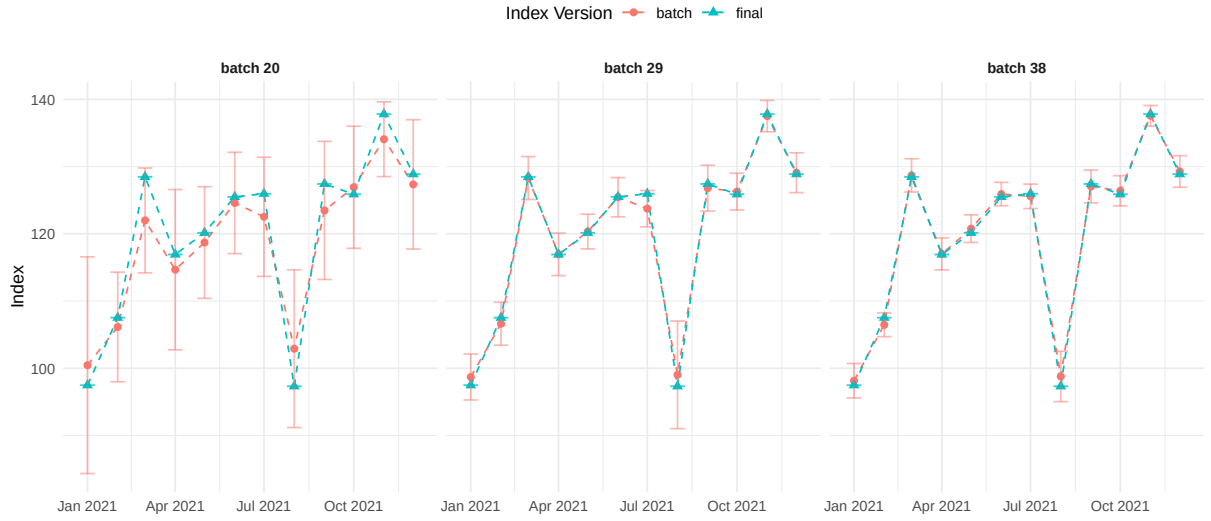

Figure 5.5 General Index Series from Jan, 2021 to Dec, 2021.

In Figure 5.5, the three index versions (initial, each of the three batches, final) are shown with the uncertainty intervals for the prediction of the model from January 2021 to December 2021. In Figures 5.6 and 5.7, the corresponding annual and monthly variation rates are represented, for these same time periods.

The index charts clearly illustrate the model's convergence toward the actual values as each batch progresses. Furthermore, a consistent reduction in predictive uncertainty is evident, as reflected by the narrowing of the RMSE bands. While the monthly rates yield equivalent conclusions, the annual rates exhibit more pronounced discrepancies during the month of April. Although the estimates remain relatively accurate, this deviation is attributed to the unprecedented volatility of April 2020, the period most severely impacted by the COVID-19 pandemic.

[1] Due to significant updates in the LightGBM framework since 2022, the results presented in the original working paper are not directly comparable to the current findings even both analysis lead to the same conclusions.

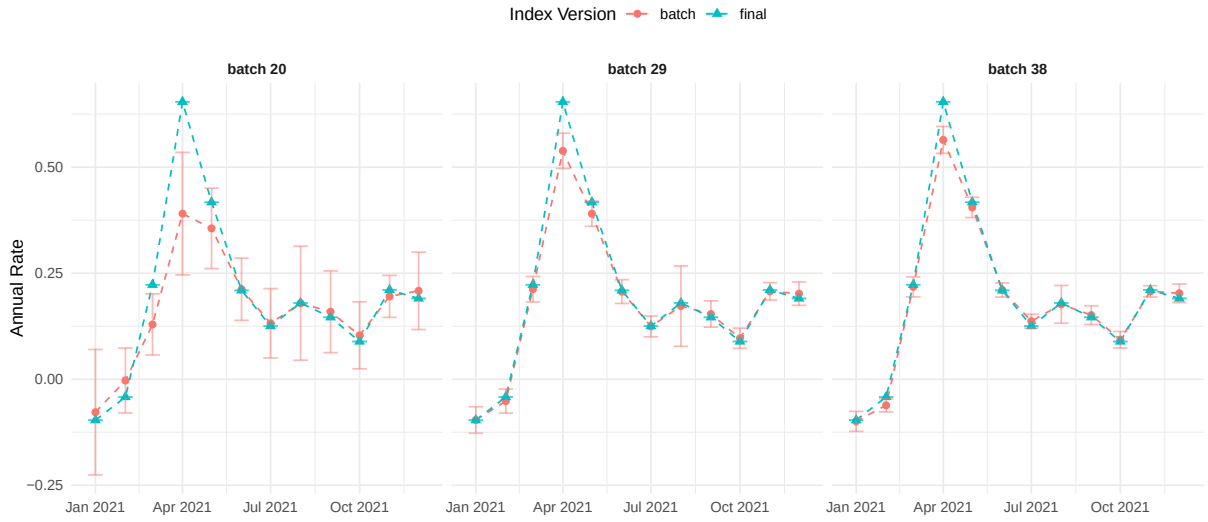

Figure 5.6 Annual Variation Rates Series from Jan, 2021 to Dec, 2021.

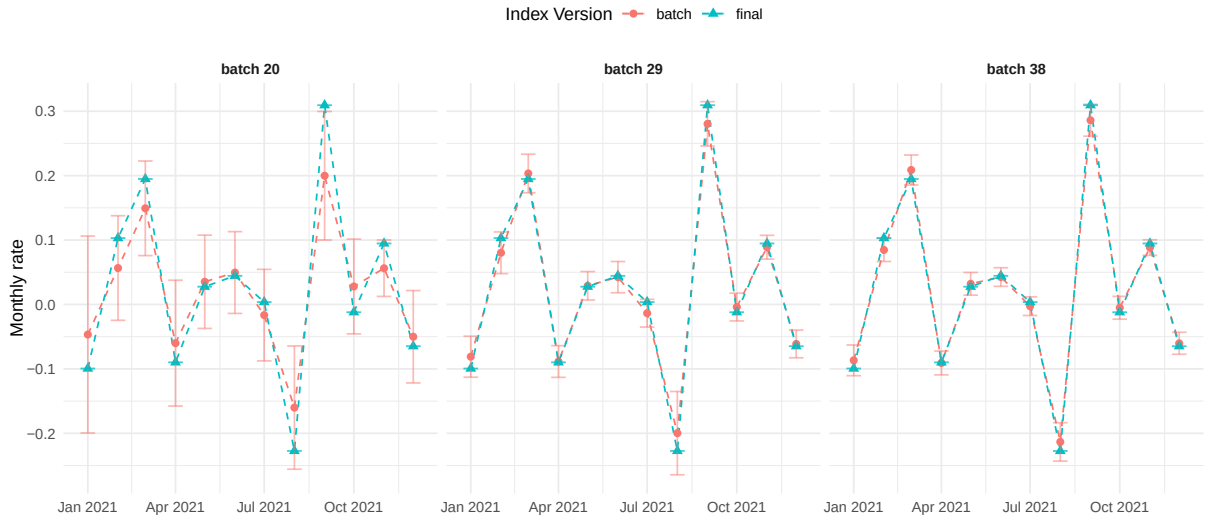

Figure 5.7 Monthly Variation Rates Series from Jan, 2021 to Dec, 2021.

Temporal cross-validation is done just before the execution of the first batch of each period, with the consequent training of the best model, then the drift of the model is avoided because the update of the model. The results about the best hyperparameters selected can be seen in Appendix in the table B.13 where it can be seen that just the `eta` parameter changes for some months. These results suggest the possibility to change the frequency of cross-validation to each six months with the advantage of saving execution time in production. The training of the model is repeated in each batch with the aim to incorporated the new collected units for the reference period.

The importance of the variables has been analyzed in all the periods. In Figure 5.8, it is shown the top 10 variables by importance measured with gain metric for December 2021. More results are shown in appendix B for the periods from January 2021 to December 2021. These results show the influence of the periodicity of the data and the special features of each month in the regressors having more impact in the predictions. Feature importance analysis reveals a preference for historical indicators. Although several key variables remain predominant

month-over-month, the model also captures seasonal dynamics and adapts its weighting in response to structural breaks or disruptive changes in economic conditions.

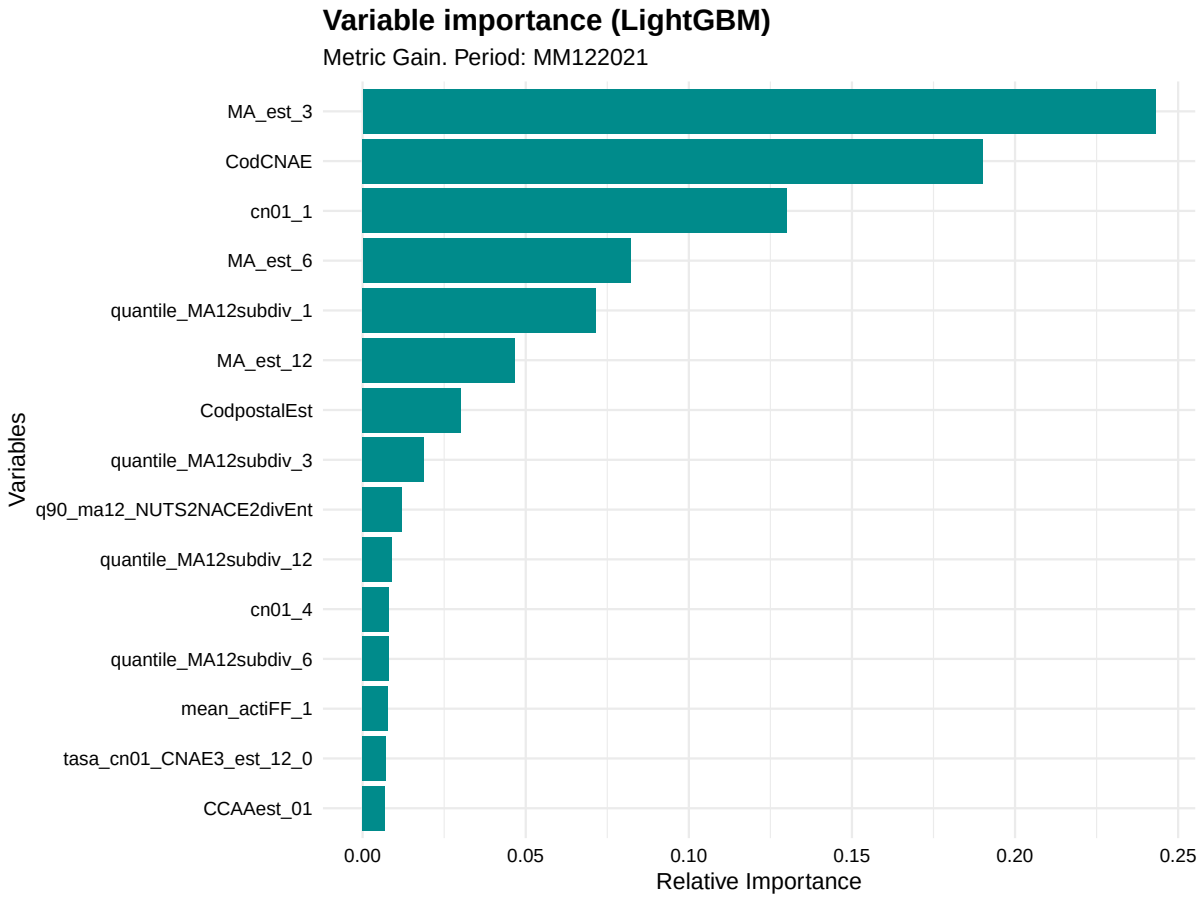

Figure 5.8 Variable importance for period Dec, 2021.

The results of the analysis of variable importance about that longitudinal data provides the most critical information is due to the extensive historical datasets generated by cut-off sampling. This kind of analysis is essential to improve the interpretability of the results. In this line, other scenarios have been evaluated, two scenarios related to different possibilities of the model and also a traditional method of imputation:

d0u: simulating the information available at $m + 1d$ (without any current regressors or reference period units), named in the graphs and tables as *d0units* or *d0u*,

d20u model0: predictions are calculated for each batch using available units but excluding current regressors, named in the graphs and tables as *d20units model0*,

mean: the mean of the same group of economic activity and stratum is imputed for the not yet collected units, named in the graphs and tables as *mean*.

As illustrated by the comparison of lines in Figure 5.9, the inclusion of data from collected units in each batch, and the subsequent model retraining, is of significantly greater importance than the marginal contribution of current regressors. The model, without the units from the reference period, fails to capture shifts in the economic environment and is excessively smooth.

Figure 5.9 General Index Series from Jan, 2021 to Dec, 2021 for the two new scenarios with the model: 1) training with all available units but without any current regressors, just with longitudinal regressors (named d20units model0); 2) training without the data collected in the reference period, just historical units so that just longitudinal regressors are available (named as d0units).

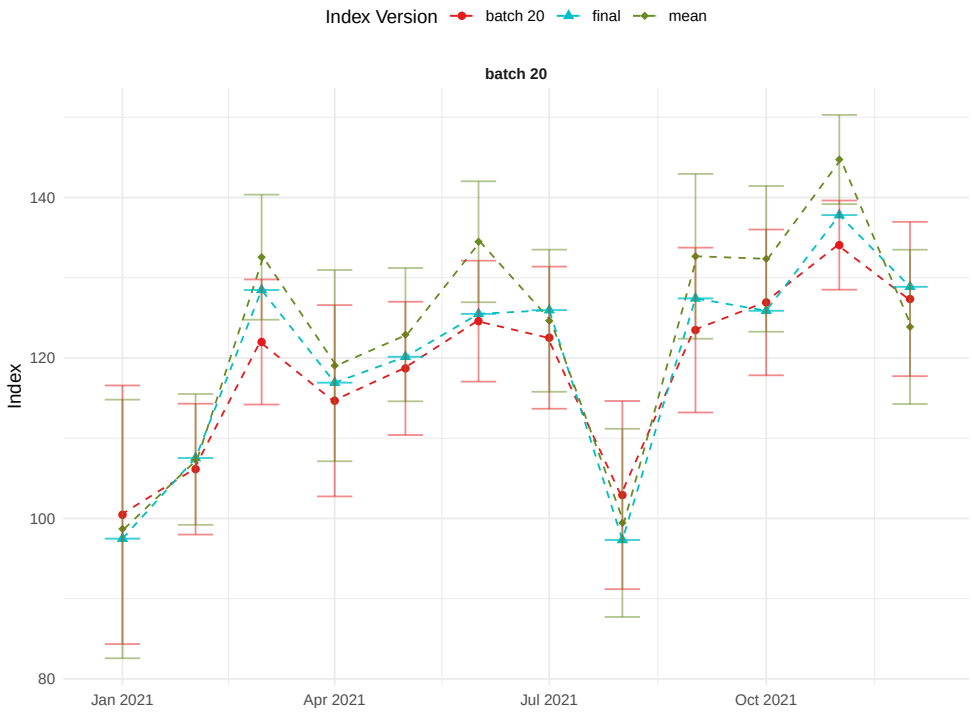

Figure 5.10 General Index Series from Jan, 2021 to Dec, 2021 for the best scenario with the model and the traditional method with the mean.

Figure 5.10 presents the aggregate-level comparison between the most comprehensive scenario, by using the model that incorporates all units and regressors, and the traditional method based on the mean by stratum. Although the differences are not particularly pronounced, the model-based scenario exhibits a more stable behavior.

Numerical comparison is more accurate and it can be seen in the table 5.2 for batch 20 ($d = 20$) with the relative differences of the indices predicted without collected units of the reference period (d0u index), in case of doing the training with the collected units (d20u index) and the traditional mean imputation (mean).

Table 5.2 Macro level comparison: indices and relative differences at $d = 20$ moment (batch 20).

Period	mean index	d0u index	d20u index	final index	mean rel.dif.	d0u rel.dif.	d20u rel.dif.
MM012021	98,6916	106,8620	100,4601	97,4833	0,0124	0,0962	0,0305
MM022021	107,3589	106,6207	106,1457	107,5283	-0,0016	-0,0084	-0,0129
MM032021	132,5606	109,6918	121,9922	128,4738	0,0318	-0,1462	-0,0505
MM042021	119,0477	112,5757	114,6679	116,9247	0,0182	-0,0372	-0,0193
MM052021	122,9041	112,1445	118,7049	120,1501	0,0229	-0,0666	-0,0120
MM062021	134,4845	118,2951	124,5887	125,4969	0,0716	-0,0574	-0,0072
MM072021	124,6468	119,9611	122,5287	125,9617	-0,0104	-0,0476	-0,0273
MM082021	99,4475	114,6682	102,9138	97,3218	0,0218	0,1782	0,0575
MM092021	132,6701	113,0402	123,4793	127,4236	0,0412	-0,1129	-0,0310
MM102021	132,3543	132,5860	126,9262	125,8839	0,0514	0,0532	0,0083
MM112021	144,7346	124,4136	134,0675	137,8039	0,0503	-0,0972	-0,0271
MM122021	123,8772	129,9876	127,3503	128,8729	-0,0388	0,0086	-0,0118

Expanding the analysis to a comparative study of the d20u and d0u scenarios for the day 20 ($m + 20d$) early estimates, the resulting RMSE values are presented in the table 5.3. A noteworthy example occurs in October 2021: while the relative difference at the macro level is significantly lower for the d20u scenario, the micro-level comparison reveals a slightly higher RMSE.

Consequently, it is evident that quality assessments must be conducted at both the micro and macro levels, with particular emphasis on the aggregate, as it constitutes the final output. However, because our approach focuses on reconstructing the microdata, the micro-level metrics must be sufficiently accurate, something the traditional mean-imputation method fails to achieve, as it yields substantially higher RMSE across all periods.

Table 5.3 Micro level comparison with the RMSE at $d = 20$ moment (batch 20).

Period	mean RMSE	d0u RMSE	d20u RMSE
MM012021	8929586,96	1625649,28	1796822,79
MM022021	9949607,32	3283048,06	2958609,20
MM032021	11172922,20	4952708,21	3228486,20
MM042021	7240216,57	1906268,75	1863876,43
MM052021	6344414,12	1904695,41	1928537,31
MM062021	7164319,36	3719408,01	3841446,39
MM072021	8875326,47	3248044,50	2502421,65
MM082021	6181275,83	2973956,97	2362758,28
MM092021	7970286,61	2369746,69	2033511,62
MM102021	7641160,44	1365172,33	1398078,17
MM112021	8612556,60	2072884,37	1871149,92
MM122021	13481986,10	3765284,30	3598944,80

In conclusion, the most significant gains in prediction precision and aggregate accuracy are derived from the model scenario where units collected within the reference period are incorporated to the training of the model.

5.4. Early imputation in accommodation establishments (PL)

This section presents the results obtained from the best-performing model selected in the previous stage of the analysis. To evaluate its accuracy and its ability to reproduce the observed seasonal structure, the predicted values are compared with the actual totals, both in aggregated seasonal terms and in a more detailed time perspective. The comparison is first illustrated graphically in Figure 5.11, which presents the dynamics of actual and predicted totals over time.

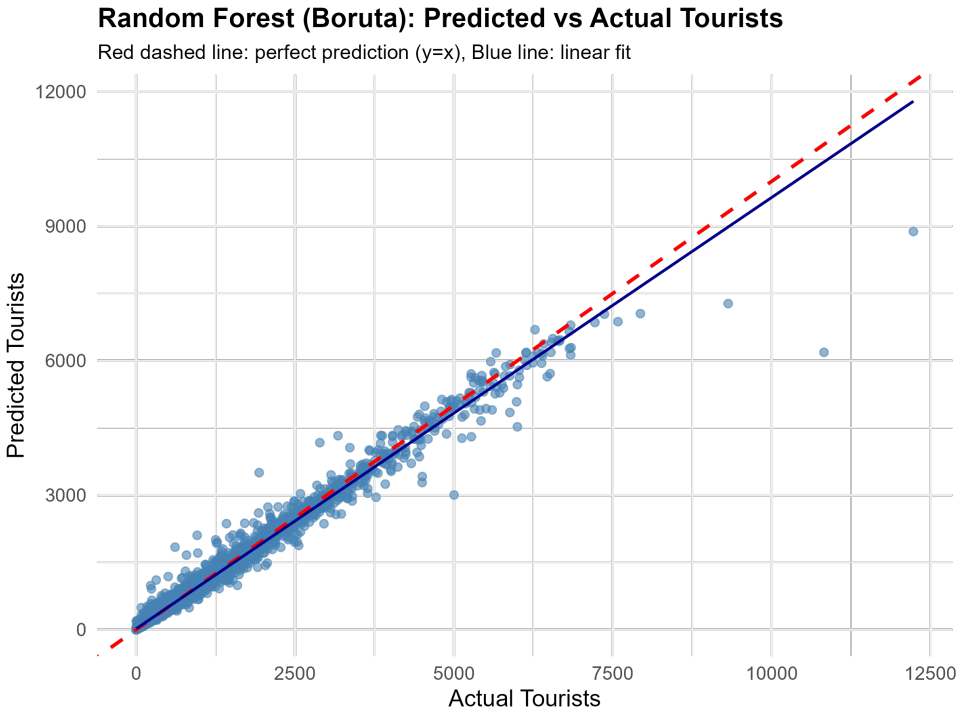

Figure 5.11 Actual and predicted totals for the best-performing model

The table shows the actual and predicted totals for each season. In autumn, the predicted value of 1,590,985 slightly overestimates the actual total of 1,563,664 by about 27,321, which is roughly 1.7% higher. In summer, the prediction of 2,852,318 slightly underestimates the actual value of 2,886,669 by 34,351, or about 1.2% lower. For spring, the predicted value of 814,434 is slightly below the actual total of 823,226, a difference of 8,792, approximately 1.1% lower. Finally, in winter, the prediction of 839,577 exceeds the actual total of 811,132 by 28,445, which corresponds to about 3.5% higher.

Season	Total Actual	Total Predicted
Autumn	1,563,664	1,590,985
Summer	2,886,669	2,852,318
Spring	823,226	814,434
Winter	811,132	839,577

Table 5.4 Actual vs Predicted Totals by Season

Overall, the predictions are quite close to the actual values, with only minor seasonal deviations. The largest relative error occurs in winter. These results suggest that the model captures the seasonal pattern reasonably well, although it slightly overestimates totals in autumn and winter, while slightly underestimating totals in spring and summer.

Breaking the comparison down month by month over the analyzed period, we can see that the model generally follows the trend of the actual totals, capturing the seasonal peaks and troughs fairly well. From April 2018 to March 2019, predictions closely track the actual values, with minor deviations. For instance, in April 2018, the prediction of 118,223 slightly underestimates the actual total of 120,602, while in May 2018, the predicted 160,861 is below the actual 167,271. During the summer months, such as July and August 2018, the model captures the seasonal peak accurately, with predicted values of 455,308 and 536,485 compared to actuals of 459,517 and 540,068, respectively. The fall and winter months show slightly higher deviations, for example, in November and December 2018, the predicted totals exceed the actuals by roughly 6,000–8,000, indicating a tendency to overestimate in colder months.

In 2019, the predictions continue to follow the overall seasonal pattern. Spring months like April to June 2019 show small underestimations, while the summer peak in July and August 2019 is again captured well, though August 2019 shows a noticeable underprediction of 654,202 compared to the actual 673,195. Fall and winter months generally display modest overestimation, similar to the previous year.

The year 2020 presents a very different pattern, likely due to the disruptions caused by external factors, such as the COVID-19 pandemic. Early months like January and February 2020 remain relatively close to actual values. However, in March 2020, the actual total drops sharply to 40,046 while the prediction is 53,474, an overestimation of roughly 33%. The following months, especially May and June 2020, show extremely low actual totals (e.g., 895 in May) with predictions still slightly higher, indicating that the model struggles to fully capture abrupt drops. From July to December 2020, predictions align more closely with reality again, although deviations remain noticeable in the lowest months, such as November and December, where predictions exceed actual totals by a few thousand.

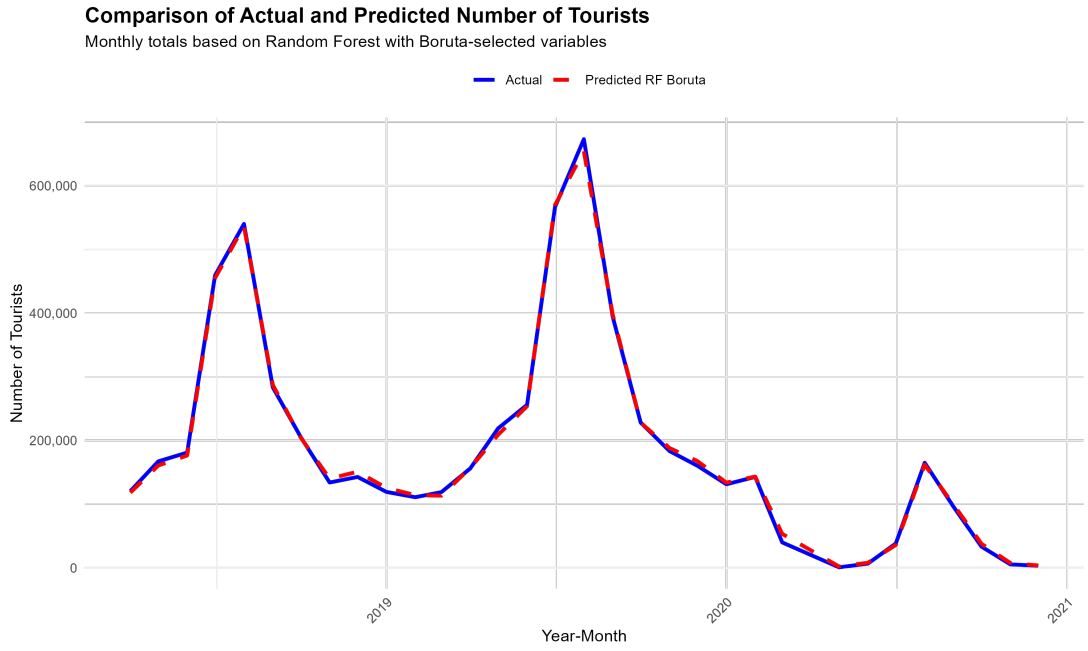

Figure 5.12 Comparison of Actual and Predicted Totals Over Time

Overall, this monthly analysis highlights that the model is quite effective in capturing seasonal patterns and the general trend over multiple years. As Figure 5.12 shows, its performance

is strongest during stable periods, especially summer peaks, while it tends to overestimate during sudden drops caused by extraordinary events and slightly overpredicts in winter months. This suggests that the model is robust for regular seasonal forecasting but may require additional adjustment to handle extreme events or sudden anomalies.

The table 5.5 presents the actual and predicted number of tourists for each voivodeship. Overall, the Random Forest model with Boruta-selected variables performs reasonably well in capturing the observed tourist flows.

Table 5.5 Actual and Predicted Number of Tourists by Voivodeship

Voivodeship	Total Actual	Total Predicted	% Difference
dolnośląskie	742 767	753 711	1.47%
kujawsko-pomorskie	328 216	329 559	0.41%
lubelskie	97 989	98 889	0.92%
lubuskie	77 998	78 199	0.26%
łódzkie	145 574	143 989	-1.09%
małopolskie	1 723 404	1 712 216	-0.65%
mazowieckie	997 144	1 005 084	0.80%
opolskie	1 985	2 067	4.13%
podkarpackie	88 501	90 144	1.86%
podlaskie	163 872	164 741	0.53%
pomorskie	322 343	322 591	0.08%
śląskie	628 024	639 012	1.75%
świętokrzyskie	55 803	58 058	4.04%
warmińsko-mazurskie	56 547	56 091	-0.81%
wielkopolskie	140 957	142 696	1.23%
zachodniopomorskie	513 565	500 259	-2.59%

In the largest voivodeships, such as Małopolskie and Mazowieckie, the predicted numbers are very close to the actual totals, with slight underestimation in Małopolskie (1,712,216 predicted vs. 1,723,404 actual; -0.65%) and slight overestimation in Mazowieckie (1,005,084 predicted vs. 997,144 actual; 0.80%). Similarly, in Dolnośląskie, the prediction is slightly higher than the observed value (753,711 vs. 742,767; 1.47%).

For smaller voivodeships, such as Opolskie or Świętokrzyskie, the predicted values are also reasonably accurate given the low totals, although the relative differences are somewhat higher (4.13% and 4.04%, respectively), which is typical for regions with smaller absolute values. Some minor deviations are observed in Warmińsko-Mazurskie (56,091 predicted vs. 56,547 actual; -0.81%) and Zachodniopomorskie (500,259 predicted vs. 513,565 actual; -2.59%), but these differences remain modest in percentage terms.

The percentage differences confirm that, for most voivodeships, prediction errors remain within ±2%, indicating a high level of accuracy. Only a few regions exceed this range, primarily those with relatively low tourist volumes, where small absolute discrepancies translate into larger percentage differences.

Overall, the model shows good predictive performance across Poland, effectively capturing regional variations in tourist numbers while maintaining accuracy for both large and small regions.

The table 5.6 presents the actual and predicted number of tourists by type of accommodation establishment. Overall, the Random Forest model with Boruta-selected variables shows good predictive performance across different types of facilities.

Table 5.6 Actual and Predicted Number of Tourists by Type of Establishment

Type of Establishment	Total Actual	Total Predicted	% Difference
hotel	4 150 178	4 177 305	0.64%
motel	17 820	17 704	-0.65%
boarding house	78 666	78 927	0.33%
other hotel establishment	426 343	420 869	-1.29%
excursion hostel	7 864	8 663	10.15%
shelter	15 144	15 639	3.26%
youth shelter	7 703	7 445	-3.40%
school youth shelter	25 588	26 725	4.44%
holiday centre	220 137	216 226	-1.77%
holiday youth centre	4 849	4 942	1.91%
training-recreational centre	187 807	181 497	-3.36%
creative arts centre	3 728	3 765	0.99%
complex of tourist cottages	38 924	38 601	-0.83%
camping site	44 445	44 758	0.70%
tent camp sites	34 202	33 631	-1.67%
hostel	294 371	289 120	-1.78%
health establishment	205 625	211 088	2.64%
room for guest	188 162	187 112	-0.55%
agrotourism lodging	23 600	23 579	-0.09%
other tourist accommodation	109 533	109 709	0.16%

Hotels account for the largest share of tourists, with over 4.1 million actual visitors, and the model slightly overestimates this number (4,177,305 predicted, +0.64%). Other large segments include hostels and holiday centres, where predicted values are also very close to the actual totals (+0.70% for camping sites, -1.78% for hostels, -1.77% for holiday centres), indicating that the model captures the main patterns in tourist distribution.

Smaller establishments, such as excursion hostels, youth shelters, and agrotourism lodgings, show minor deviations between actual and predicted numbers, but the differences remain relatively small in absolute and relative terms. For instance, excursion hostels are slightly overestimated (+10.15%), while youth shelters are slightly underestimated (-3.40%), and agrotourism lodgings are nearly identical (-0.09%).

Among all types, the largest relative deviations occur in excursion hostels and school youth shelters (+10.15% and +4.44%, respectively), suggesting that the model may slightly overpredict niche segments with lower overall visitor numbers. Conversely, mid-sized establishments such as training-recreational centres and other hotel establishments show modest underestimation (-3.36% and -1.29%, respectively).

Overall, the model demonstrates robustness across both high-volume and low-volume accommodation types. It successfully captures the variations in tourist numbers among different establishments, providing a reliable prediction for planning, resource allocation, and policy analysis purposes.

Enforcing consistency constraints or value limits on predicted variables may yield suboptimal results. Specifically, it has been observed that re-adjusting predictions to integer values or applying upper bounds negatively impacts the precision of macro-level aggregates, suggesting that the raw predictions provide a more reliable basis for estimation.

6

Conclusions

6.1. Methodology outcomes

The research conducted under this research line of WP9 establishes a methodological framework for constructing an end-to-end prototyping system to deliver early estimates for different projects: monthly short-term business statistics (ES and DE), school enrollment (IT) and number of monthly tourists in accommodation facilities (PL). The projects in Spain (ES), Germany (DE), and Poland (PL) utilize machine learning to impute missing units during the survey collection phase, enabling the reconstruction of the complete microdataset. This methodology also accounts for scenarios where no current microdata has been collected yet, instead leveraging historical data and auxiliary information from the reference period. In contrast, the project in Italy (IT) uses the SchoolEnrollment dataset that contains only information from administrative sources from previous periods, and the aim is to provide an early prediction of the variable of interest, which becomes available from administrative sources only at a later stage.

The approach of these projects relies on different algorithms such as: random forest for classification and for regression or gradient boosting regression model. The models are trained on historical microdata, aggregated survey data, administrative registers data and partial information from units that have already responded in the ongoing reference period. The resulting reconstructed microdataset allows the computation of early estimates using the standard production process. In some cases they can even be complemented with uncertainty intervals to assess reliability.

The methodological design of the machine learning process emphasises modularity, enabling incremental improvements along all phases of the statistical learning pipeline.

Key Learning Points and Methodological Insights:

- **No "One-Size-Fits-All" Methodology:** Evidence from Italy, Spain and Poland demonstrated that Machine Learning models (Random Forest or gradient boosting) can significantly outperform traditional techniques. Conversely, the German case showed that linear regression remains more robust and maintainable when a strong linear relationship is inherent in the data. This highlights that **model selection must be driven by the underlying data structure and patterns** rather than the sheer complexity of the algorithm.

- **The Paramount Importance of Real-Time Data:** The Spanish pilot study provided a crucial lesson: retraining models with newly collected units from the reference period has a profound impact on precision. This approach goes beyond merely mitigating model drift; it ensures the system is **responsive enough to detect sudden shifts in behavioral**

patterns, a capability that is vital for the relevance and accuracy of short-term economic surveys.

- **Balancing Micro and Macro Accuracy:** Results from Italy and Spain underscore that optimizing for individual-level (micro) precision can, at times, compromise the quality of the aggregate (macro) distribution. Careful hyperparameter tuning with a macro-level metric can be a **balance mechanism to ensure statistical coherence**. Ultimately, this reinforces that aggregate accuracy must remain the overarching objective of the statistical process.

- **Navigating the "Timeliness vs. Accuracy" Trade-off:** In both Germany and Spain, predictive accuracy improved drastically as more days passed following the close of the reference month. For instance, the German experience showed that waiting from $m + 15d$ to $m + 20d$ significantly reduced error due to the increased availability of edited data. It is essential to **empirically evaluate the "tipping point"** where this trade-off is optimized for final dissemination.

- **Vulnerability to Extraordinary Shocks:** The Polish project revealed that while ML models excel at capturing complex seasonal and regional rhythms during stable periods, they are vulnerable to disruptive events like the COVID-19 pandemic. This indicates that production-grade models require **robust monitoring metrics** to facilitate expert-led supervision and manual intervention when anomalies arise. In contrast, the Spanish project mitigates this vulnerability by implementing tailored feature engineering. By constructing regressors specifically designed for outliers and evolving patterns, the model's adaptability to disruptive events is significantly improved.

- **Operational Stability and Efficiency:** In the Spanish case, the consistency of hyperparameters over several months suggests that exhaustive monthly cross-validation may be unnecessary. **Streamlining the frequency of model tuning** can yield significant computational savings in real-world production environments without sacrificing the quality or reliability of the final estimates.

- **Interpretability:** The Spanish project integrates interpretability to provide a deeper understanding of its outcomes. This is accomplished by analyzing variable importance and performing comparative studies across various scenarios.

Although significant progress has been made, the following components remain open for improvement:

- expansion of regressor sets,

- optimization of hyperparameters and the systematic comparison of alternative statistical learning algorithms,

- explicit modelling of measurement errors through complementary models,

- a systematic study on model variable importance is needed to provide interpretability of the early estimates so that we can understand how regressors affect the predictions,

- indicators or metrics to keep under control quality in coherence with the total machine learning error model,

- robustness evaluation of the different models.

Despite these potential enhancements, the current feasibility assessment already yields reliable early estimates for the ES-ITI, DE-ITI, and PL-Accommodation projects. These results provide an evidence-based foundation for both methodological development and the transition to the next phase: implementing a prototype ready to be deployed in production. Regarding the IT-SchoolEnrollment project, even with the optimal Random Forest model, micro-level quality remains insufficient for deployment. Consequently, future work will explore alternative approaches, such as Neural Networks with embedding layers, to better leverage information like municipalities and school codes.

These initiatives demonstrate that incorporating machine learning tools into official statistics does not diminish the role of statistical officers; rather, it transforms it. As statistical processes become more automated, human intervention shifts from executing manual tasks to designing, supervising, and improving algorithmic components. Machine learning-based imputation continues to require substantial human intervention for the identification and validation of regressors in the feature engineering phase. This evolution requires reconsideration of traditional editing and imputation workflows, potentially leading to new or redefined business functions within statistical production that are described in GSDEM. Moreover, the integration of subject matter expertise is not merely advisable but requisite to ensure that predictive models align with the high standards of quality and interpretability inherent to official statistics. Their involvement remains essential to ensure the quality and interpretability of the resulting predictive models.

6.2. Production focused strategy

The adoption of a modular architecture is foundational for the transition from experimental prototypes to robust production environments. By aligning with international standards such as the GSBPM, this framework facilitates the scalability and reusability of statistical learning pipelines across diverse domains and national offices. Currently, a significant challenge remains in formally mapping algorithmic pipelines with Machine Learning techniques to GSBPM activities. Such an end-to-end automated approach allows for the processing of high-frequency data, significantly enhancing the timeliness and accuracy of early estimates. Moreover, this architecture enables the integration of complementary models, such as those predicting measurement errors, to mitigate bias and refine the quality of the statistical output in real-world production settings.

To ensure the successful transition of methodological innovations into production, the following issues must be addressed:

Methodological Issues

- **GSBPM Mapping:** Establishing a formal framework to align statistical learning pipelines with international production standards.
- **Evolution of Business Functions:** Rethinking traditional editing and imputation strategies to accommodate algorithmic automation and new validation requirements. As it is shown in this report, new techniques such as ML can help in the use of imputation with new goals as early imputation.
- **Quality Assurance:** Implementing complementary models with some additional information to detect measurement errors and ensure the interpretability of predictive models, ensuring that quality assessment is rigorously aligned with the *Total Machine Learning Error* (TMLE) framework to provide a holistic evaluation of model performance.

Computational Issues

- **High-Performance Infrastructure:** Establishing a robust computational architecture with sufficient capacity to support the intensive demands of machine learning workflows, enabling the use of incremental computation of aggregates and parallel evaluation across multiple levels to handle large-scale datasets efficiently.
- **MLOps Framework and Orchestration:** Adopting MLOps (Machine Learning Operations) paradigms to organize and automate the end-to-end lifecycle. This includes version control for models and data, continuous integration and deployment (CI/CD), and rigorous pipeline monitoring to ensure the stability and reproducibility of the results if it is allowed by the nature of the algorithm; or reproducibility of the process to different applications.
- **Streaming Architectures:** Implementing specialized streaming algorithms for real-time evaluation and process automation, allowing for the continuous ingestion and processing of data during the collection phase.
- **Advanced Diagnostic Visualization:** Developing multi-level dashboards and aggregation error heatmaps to monitor micro and macro metrics, providing clear insights into error decomposition and model performance across different domains.

Organizational Issues

- **Interdisciplinary Collaboration:** Strengthening the synergy between methodological units and IT experts to ensure seamless deployment in production environments. Integration of subject matter expertise is essential to ensure the quality and interpretability of the resulting predictive models.
- **Human-in-the-loop Paradigms:** Adjusting the role of statistical officers from manual processing to the strategic supervision and validation of automated systems.
- **Knowledge Transfer:** Developing shared repositories and modular tasks that can be reused across different surveys and national statistical institutions.

The second task of WP9 focuses on developing a production-ready prototype for those projects that demonstrate both viability and methodological maturity. In this regard, the initiatives from Germany, Spain, and Poland have expressed their readiness to move forward into the production-preparation phase.

6.3. Results as input of other work packages

The insights and outcomes of this report serve as valuable input for other work packages within the AIML4OS project. In particular, the chapters 3 and 4 are explained with enough generality of the methodology developments that can be useful for WP5 (Standards, methodological and implementation frameworks) and WP6 (Knowledge repository and training materials). Some ideas that could be useful are shown in the following.

WP5 inputs:

- Within the TMLE framework, the core components are clearly reflected in the projects presented; a notable example is the direct correspondence between cross-validation procedures and the principle of internal validity.
- The diverse range of project designs demonstrates the wide applicability of these models across different sampling strategies, including both probability sampling and cut-off designs.

- Impact on Data Quality: Implementing these machine learning techniques for early imputation significantly enhances the timeliness dimension of data quality. This document demonstrates that the findings highlight the potential for daily model training and high-frequency estimates, offering a foundation for real-time statistical production and the adaptation of collection systems to support automated, ML-driven pipelines.
- Transparency and Interpretability: Furthermore, the systematic analysis of variable importance provides critical insights that increase model transparency, align with quality frameworks, and improve the overall interpretability of early estimates.

WP6 inputs:

- A document of training material will be developed with the methodology outcomes of this report about feature engineering.

6.4. Final remarks

Statistical learning techniques are poised to become a versatile and powerful tool for producers of official statistics, enabling continuous quality improvement in both timeliness and accuracy. This pilot study provides concrete evidence that early estimates can be produced with reasonable reliability using machine learning–based imputation applied to partially collected survey microdata. Four projects with categorical and numerical data as target and different subject matters have been carried out with successful results.

Further research is still needed in several areas, including the modelling of measurement error, adapting the process to be automated, integrating daily data streams, developing robust methods for model interpretability and developing new indicators and metrics to keep quality under control from the input to the output.

Machine learning does not displace the statistician; it empowers the expert. This work proves that advanced algorithms, when guided by human oversight, significantly amplify the capacity and precision of official statistics. The experience gained here stands as a pivotal milestone in our transition toward a modernized production framework, where responsible innovation and subject matter expertise converge with a basis of methodology to define the future of official statistics.

Bibliography

K. P. Adithya and P. Ancy. A survey on machine learning based imputation techniques for missing data. In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1415–1421. IEEE, 2022. doi: 10.1109/ICCMC53470.2022.9762231. URL https://ieeexplore.ieee.org/document/9762231.

Mustafa Alabadla, Fatimah Sidi, Iskandar Ishak, Hamidah Ibrahim, Lilly Suriani Affendey, Zafienas Che Ani, Marzanah A. Jabar, Umar Ali Bukar, Navin Kumar Devaraj, Ahmad Sobri Muda, Anas Tharek, Noritah Omar, and M. Izham Mohd Jaya. Systematic review of using machine learning in imputing missing values. *IEEE Access*, 10:44483–44502, 2022. doi: 10.1109/ACCESS.2022.3160841. URL https://ieeexplore.ieee.org/document/9762231.

S Barragán, L Barreñada, JF Calatrava, JC Gálvez Sáenz de Cueto, JM Martín del Moral, E Rosa-Pérez, and D Salgado. Early estimates of the industrial turnover index using statistical learning algorithms. *Statistics Spain Working Paper*, 3:22, 2022. URL https://www.ine.es/GS_FILES/DocTrabajo/art_doctr032022.pdf.

C. Bentéjac, A. Csörgő, and G. Martínez-Mu noz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021. doi: 10.1007/s10462-020-09896-5.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012. URL https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

B. Bok, D. Caratelli, D. Giannone, A. Sbordone, and A. Tambalotti. Macroeconomic nowcasting and forecasting with Big Data. Technical report, Staff Report, No. 830, Federal Reserve Bank of New York, 2017. URL https://www.newyorkfed.org/medialibrary/media/research/staff_reports/sr830.pdf. https://www.econstor.eu/handle/10419/189871.

Brandyn Bok, Daniele Caratelli, Domenico Giannone, Argia M. Sbordone, and Andrea Tambalotti. Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics*, 10:615–643, 2018. doi: 10.1146/annurev-economics-080217-053214.

George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001. URL https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf.

Sofie De Broe, Olav ten Bosch, Piet Daas, Gert Buiten, Ben Laevens, and Bert Kroese. The need for timely official statistics. The COVID-19 pandemic as a driver for innovation. *Statistical Journal of the IAOS*, 37:1221–1227, 2021. doi: 10.3233/SJI-210825. URL https://journals.sagepub.com/doi/10.3233/SJI-210825.

Andriy Burkov. *Machine Learning Engineering*. True Positive Incorporated, 2020. ISBN 978-1999579579. URL http://www.mlebook.com.

R.L. Chambers. Outlier Robust Finite Population Estimation. *Journal of the American Statistical Association*, 81(396):1063–1069, 1986. doi: 10.1080/01621459.1986.10478374. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478374.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chao Chen, Andy Liaw, and Leo Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12):24, 2004. URL https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.

François Chollet and J.J. Allaire. *Deep Learning with R*. Manning Publications, Shelter Island, NY, 2018. URL https://www.manning.com/books/deep-learning-with-r.

D. D. Lazer, R. Kennedy, G. King, and A. Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science*, 343:1203–1205, 2014. URL https://gking.harvard.edu/sites/g/files/omnuum7116/files/gking/files/0314policyforumff.pdf.

M. Dagdoug, C., Goga, and D. Haziza. Imputation Procedures in Surveys Using Nonparametric and Machine Learning Methods: an Empirical Comparison. *Journal of Survey Statistics and Methodology*, 09 2021. doi: 10.1093/jssam/smab004. URL https://doi.org/10.1093/jssam/smab004.

T. de Waal, J. Pannekoek, and S. Scholtus. *Handbook of statistical data editing and imputation*. Wiley, 2011. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9780470904848.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012. URL https://dl.acm.org/doi/10.1145/2347736.2347755.

A. Dorville et al. Towards more timely measures of labour productivity growth. Oecd statistics working papers, OECD Publishing, 2025. URL https://oecdstatistics.blog/2025/03/31/nowcasting-labour-productivity-growth/.

Florian Dumpert. WP1 - Theme 2: Edit and Imputation Report. Hlg-mos machine learning project report, United Nations Economic Commission for Europe (UNECE), 2020. URL https://statswiki.unece.org/spaces/ML/pages/290358735/WP1+-+Theme+2+Edit+and+Imputation+Report. Part of the Machine Learning for Official Statistics Project.

Florian Dumpert. *Foundations and Advances of Machine Learning in Official Statistics*. Springer, 2025. URL https://link.springer.com/book/10.1007/978-3-032-10004-7.

Tarekegn Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabiso Semong, Budzani Mphago, and Onalethata Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8 (1):140, 2021. doi: 10.1186/s40537-021-00516-9. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00516-9.

ESS. ESS Handbook for Quality Reports, 2014. URL https://ec.europa.eu/eurostat/documents/3859598/6651706/KS-GQ-15-003-EN-N.pdf/18dd4bf0-8de6-4f3f-9adb-fab92db1a568.

Eurostat. MEMOBUST handbook: Methodology of modern business statistics. Module: Imputation. Technical report, European Commission, 2014. URL https://tkbm-test.scb.se/section/imputation/. Available at the ESS-net portal for Statistical Methodology.

Eurostat. *Handbook on Rapid Estimates*, 2017. URL https://ec.europa.eu/eurostat/documents/3859598/8555708/KS-GQ-17-008-EN-N.pdf. Publications Office of the European Union, Luxembourg.

Eurostat. *Handbook on Methodology of Modern Business Statistics*, 2018. Publications Office of the European Union, Luxembourg.

Eurostat. Ess statistical production reference architecture. https://interoperable-europe.ec.europa.eu/collection/statistical-enterprise-architecture/document/ess-statistical-production-reference-architecture, 2019a.

Eurostat. Regulation (eu) 2019/2152 of the european parliament and of the council on european business statistics. Technical report, Eurostat, 2019b. URL https://eur-lex.europa.eu/eli/reg/2019/2152/oj/eng.

Eurostat. European business statistics regulation. commission implementing regulation 2020/1197. Technical report, 2020. URL https://eur-lex.europa.eu/eli/reg_impl/2020/1197/oj/eng.

Eurostat. *ESS Handbook for Quality and Metadata Reports*. Publications Office of the European Union, Luxembourg, 2021. doi: 10.2785/112613. URL https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/ks-gq-21-021. Detailed discussion on the trade-off between timeliness and accuracy.

Eurostat. GDP nowcasting for the euro area and member countries – 2025 edition. Statistical working papers, European Commission, 2025. URL https://ec.europa.eu/eurostat/web/products-statistical-working-papers/w/ks-01-25-042.

J.H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29:1189–1232, 2001.

Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014. URL https://dl.acm.org/doi/10.1145/2523813.

Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 3rd edition, 2022. Chapter 2: End-to-End Machine Learning Project - Create a Test Set.

D. Giannone, L. Reichlin, M. Bańbura, and M. Modugno. Now-casting and the real-time data flow. In G. Elliott and A. Timmermann, editors, *Handbook of Economic Nowcasting*, chapter 4, pages 195–237. Elsevier, Amsterdam, 2013. URL https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp1564.pdf.

L. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

Robert M. Groves and Lars Lyberg. Total survey error: Past, present, and future. *Public Opinion Quarterly*, 74(5):849–879, 01 2010. ISSN 0033-362X. doi: 10.1093/poq/nfq065. URL https://academic.oup.com/poq/article/74/5/849/1817502.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

D.J. Hand. Statistical challenges of administrative and transaction data. *Journal of the Royal Statistical Society A*, 181:555–605, 2018. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssa.12315.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328, 2008.

High-Level Group for the Modernisation of Official Statistics (HLG-MOS). The use of machine learning in official statistics. Technical report, United Nations Economic Commission for Europe (UNECE), July 2024. URL https://unece.org/sites/default/files/2024-07/HLGMOS%20The%20use%20of%20machine%20learning%20in%20official%20statistics.pdf. Modernisation of Official Statistics Series.

Tim Holt. Official statistics, public policy and public trust. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 171(2):323–346, 2008. URL https://doi.org/10.1111/j.1467-985X.2007.00523.x.

Mia Hubert and Eva Vandervieren. An Adjusted Boxplot for Skewed Distributions. *Computational Statistics & Data Analysis*, 52:5186–5201, 2008.

Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

INE. *Industrial Turnover Indices & Industrial New Orders Received Indices. Base 2021*, 2024. URL https://www.ine.es/metodologia/t05/t0530053_2021.pdf.

International Monetary Fund. *Data Quality Assessment Framework (DQAF) for Principal Macroeconomic Datasets*. IMF Statistics Department, 2012. URL https://dsbb.imf.org/dqrs/DQAF. Focuses on Serviceability and Timeliness as key dimensions of quality.

Amaia Iparragirre, Thomas Lumley, Irantzu Barrio, and Inmaculada Arostegui. Variable selection with LASSO regression for complex survey data. *Stat*, 12(1):e578, 2023. doi: 10.1002/sta4.578. URL https://onlinelibrary.wiley.com/doi/full/10.1002/sta4.578.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 112. Springer, 2013. URL https://www.statlearning.com/.

Anurag Kanaujia and Deepika Yadav. Feature engineering in machine learning. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 749–755. IEEE, 2015.

Dennis Kant, Andreas Pick, and Jasper de Winter. Nowcasting GDP using machine learning methods. *Journal of Applied Econometrics (forthcoming / Working Paper version)*, 2024. URL https://link.springer.com/article/10.1007/s10182-024-00515-0.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012. URL https://dl.acm.org/doi/10.1145/2382577.2382579.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, 30:3149–3157, 2017. NIPS 2017.

Azal Ahmad Khan, Omkar Chaudhari, and Rohitash Chandra. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Systems with Applications*, 244:122778, 2024. doi: 10.1016/j.eswa.2023.122778. URL https://doi.org/10.1016/j.eswa.2023.122778.

R. Kitchin. Big Data and Official Statistics: Opportunities, challenges and risks. *Statistical Journal of the IAOS*, 31:471–481, 2015. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2595075.

Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial Intelligence*, 97 (1-2):273–324, 1997.

Max Kuhn. *caret: Classification and Regression Training*, 2024. URL https://CRAN.R-project.org/package=caret. R package version 6.0-94.

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer, 2013. URL https://link.springer.com/book/10.1007/978-1-4614-6849-3.

R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley, Hoboken, 2nd edition, 2002.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

Aleša Lotrič Dolinar, Sašo Polanec, and Mojca Bavdaž. Real-time economic indicators in NSIs. *Statistical Journal of the IAOS*, 41(2):294–304, 2025. doi: 10.1177/18747655251341466. URL https://doi.org/10.1177/18747655251341466.

Orietta Luzi, Jeroen Pannekoek, Ton De Waal, Susan Edmonds, Pedro Revilla, Roxane Silberman, and Paula Vicard. Edimbus: Recommended practices for editing and imputation in cross-sectional business surveys. Methodologies and working papers, Eurostat, Luxembourg, 2007. URL https://ec.europa.eu/eurostat/documents/64157/4374310/30-Recommended+Practices-for-editing-and-imputation-in-cross-sectional-business-surveys-2008.pdf. Produced by the EDIMBUS project.

Nathaniel MacNell, Lydia Feinstein, Jesse Wilkerson, Päivi M Salo, Samantha A Molsberry, Michael B Fessler, Peter S Thorne, Alison A Motsinger-Reif, and Darryl C Zeldin. Implementing machine learning methods with complex survey data: Lessons learned on the impacts of accounting sampling weights in gradient boosting. *PLOS ONE*, 18(1):e0280387, 2023. doi: 10.1371/journal.pone.0280387. URL https://doi.org/10.1371/journal.pone.0280387.

Michael Mayer. *missRanger: Fast Imputation of Missing Values*, 2023. URL https://CRAN.R-project.org/package=missRanger. R package version 2.5.0.

Robert McGill, John W Tukey, and Wayne A Larsen. Variations of box plots. *The American Statistician*, 32:12–16, 1978.

Microsoft Corporation. Lightgbm, 2022. URL https://lightgbm.readthedocs.io/en/latest/index.html.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. URL https://moodle2.units.it/pluginfile.php/241303/mod_resource/content/1/Murphy_Machine_Learning.pdf.

Malte Nalenz, Julian Rodemann, and Thomas Augustin. Learning de-biased regression trees and forests from complex samples. *Machine Learning*, 113(6):3379–3398, 2024. doi: 10.1007/s10994-023-06439-1. URL https://doi.org/10.1007/s10994-023-06439-1.

OECD. Nowcasting Trade in Value Added (TiVA) indicators. Technical report, United Nations Economic Commission for Europe (UNECE), May 2023. URL https://www.oecd.org/en/publications/nowcasting-trade-in-value-added-indicators_00f8aff7-en.html. Presented at the Joint UNECE/OECD Workshop on Economic Statistics.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Sasmita Prusty, Satyabrata Patnaik, and S. K. Dash. Survey on machine learning based data imputation techniques. In *Proceedings of the 6th International Conference on Computing and Information Processing*, ICCIP '20, pages 150–159, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3411408.3411465. URL https://doi.org/10.1145/3411408.3411465.

Marco J. H. Puts, David Salgado, and Piet J. H. Daas. Leveraging machine learning for official statistics. In Florian Dumpert, editor, *Foundations and Advances of Machine Learning in Official Statistics*, Society, Environment and Statistics, pages 15–47. Springer Nature Switzerland, Cham, Switzerland, 2025. ISBN 978-3-032-10004-7. doi: 10.1007/978-3-032-10004-7_2. URL https://doi.org/10.1007/978-3-032-10004-7_2.

Peter J Rousseeuw and Christophe Croux. Alternatives to the Median Absolute Deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993. doi: 10.1080/01621459.1993.10476408.

Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

D. Salgado and B. Oancea. On new data sources for the production of official statistics, 2020. URL https://www.ine.es/GS_FILES/DocTrabajo/art_doctr012020.pdf. arXiv:2003.06797v1.

Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model Assisted Survey Sampling*. Springer Series in Statistics. Springer-Verlag, New York, 1992. ISBN 978-0-387-40620-6. doi: 10.1007/978-1-4612-4378-6.

Yu Shi, Guolin Ke, Damien Soukhavong, James Lamb, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, and Nikita Titov. *lightgbm: Light Gradient Boosting Machine*, 2021. URL https://CRAN.R-project.org/package=lightgbm. R package version 3.3.1.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012. URL https://papers.nips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

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

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

UNECE. Generic activity model for statistical organizations v1.2, 2019a. URL https://statswiki.unece.org/display/GAMSO/GAMSO+v1.2.

UNECE. Generic statistical information model v1.2, 2019b. URL https://statswiki.unece.org/display/gsim.

UNECE. Generic statistical data editing model v2.0, 2019c. URL https://statswiki.unece.org/display/sde/GSDEM.

UNECE. High-level group for the modernisation of statistical production and services, 2021a. https://unece.org/statistics/networks-of-experts/high-level-group-modernisation-statistical-production-and-services.

UNECE. Machine learning for official statistics. Technical Report ECE/CES/STAT/2021/6, United Nations Economic Commission for Europe, Geneva, 2021b. URL https://unece.org/sites/default/files/2022-09/ECECESSTAT20216.pdf.

UNECE. Generic Statistical Business Process Model v5.2, 2025. URL https://unece.github.io/GSBPM-5.2/.

UNECE Machine Learning Project Team. Hlg-mos machine learning project final report. Technical report, United Nations Economic Commission for Europe (UNECE), December 2020. URL https://statswiki.unece.org/display/ML/HLG-MOS+Machine+Learning+Project. Presented to the Workshop on the Modernisation of Official Statistics 2020.

United Nations Economic Commission for Europe (UNECE). *Statistical Data Editing: Impact on Data Quality*, 2006. URL https://unece.org/DAM/stats/publications/editing/SDE3.pdf. Geneva.

Stef Van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press, 2nd edition, 2018. ISBN 978-1138588318. doi: 10.1201/9781315116198. URL https://stefvanbuuren.name/fimd/.

Stef van Buuren. *Flexible Imputation of Missing Data*. CRC press, second edition, 2023. URL https://stefvanbuuren.name/fimd/.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011. doi: 10.18637/jss.v045.i03. URL https://www.jstatsoft.org/article/view/v045i03.

Y. Wand and R.Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39:86–95, 1996. URL https://dl.acm.org/doi/10.1145/240455.240479.

Jeremy Watt, Reza Borhani, and Aggelos K. Katsaggelos. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge University Press, 2nd edition, 2020. doi: 10.1017/9781108690935. URL https://www.cambridge.org/highereducation/books/machine-learning-refined/0A64B2370C2F7CE3ACF535835E9D7955.

Marvin N. Wright and Inke R. König. Splitting on categorical predictors in random forests. *PeerJ*, 7:e6339, 2019. doi: 10.7717/peerj.6339. URL https://doi.org/10.7717/peerj.6339.

Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017. doi: 10.18637/jss.v077.i01. URL https://www.jstatsoft.org/article/view/v077i01.

Edesa Yadegar, Kerstin Lange, Bogdan Levagin, and Bayram Oruc. t+ 20–ein projekt zur schnellschätzung von konjunkturindikatoren. *WISTA-Wirtschaft und Statistik*, 77(3):57–73, 2025.

Wesley Yung, Siu-Ming Tam, Bart Buelens, Hugh Chipman, Florian Dumpert, Gabriele Ascari, Fabiana Rocci, Joep Burger, and InKyung Choi. A quality framework for statistical algorithms. *Statistical Journal of the IAOS*, 38(1):291–308, 2022. doi: 10.3233/SJI-210875. URL https://doi.org/10.3233/SJI-210875.

All web links and URLs cited in this bibliography were accessed and verified for availability as of March, 2026.

Appendix A

Appendix: regressors description

A.1. Early imputation in school enrollment (IT)

Table A.1 Early estimation of school enrollment (IT) regressors

Name	Type	Source	Dim.	Formula	Brief description
ETA22	num	original-admin			Age in reference year $t = 2022$
SESSO	chr	original-admin	2		Gender
COD_CITTADINANZA	chr	original-admin	173		Citizenship in reference year
FL_ITA	chr	derived-admin	2	Citizenship coded 0/1 (not Italian/Italian)	Identifier for Italian people in reference year t
AREA_CITTADINANZA		derived-admin	11	Citizenship coded in 11 area	
REG_RES	chr	original-admin	20		Region of residence in reference year t
PRO_RES	chr	original-admin	107		Province of residence in reference year t
PROCOM_RES	chr	original-admin	≈ 8000		Municipality of residence in reference year t
Time dependent variables - training set (school year 2020/2021)					
FL_GT14_21	chr	derived-admin	2	1 if ETA22-1 > 14, 0 otherwise	Identifier for individuals aged over 14
FR21_STATO_FREQUENZA	chr	original-admin	3		School enrollment status
FR21_CODICE_ISTITUTO	chr	original-admin	> 10.000		School identification code
FR21_FL_CAMBIO_IST_AP		derived-admin	2	1 if inst20 $\neq$ inst21, 0 otherwise	Changed schools from one year to the next
FR21_FLAG_SCUOLA_DIG3	chr	derived-admin	2	substr(inst21,3,1)	Type of school: State or private
FR21_FLAG_SCUOLA_DIG34	chr	derived-admin	32	substr(inst21,3,2)	Type of institution (Lyceum, etc.)
FR21_CARCERI	chr	derived-admin	2	substr(inst21,5,1)	Indicator for prison school
FR21_SERALE	chr	derived-admin	2	substr(inst21,8,1)	Indicator for evening school
VAR21	chr	original-admin	≈ 20		Level and year of the attended school
FR21_FL_M3	chr	derived-admin	2	1 if VAR21=M-3, 0 otherwise	Final year of lower secondary school
FR21_FL_BOC_AP	chr	derived-admin	2	1 if VAR20=VAR21, 0 otherwise	Individuals who have failed in the past
FR21_FL_INTPARIM3	chr	derived-admin	2	1 if VAR21=M-3 and private school	Private paritarian school indicator
FR21_PROCOM_SCU	chr	original-admin			Municipality of the attended school
FR21_FL_STESSO_COM	chr	derived-admin	2	1 if COM_RES=SCU_COM	Enrolled in school in same municipality
Time dependent variables - test set (school year 2021/2022)					
FL_GT14_22	chr	derived-admin	2	1 if ETA22 > 14, 0 otherwise	Identifier for individuals aged over 14
FR22_STATO_FREQUENZA	chr	original-admin	3		School enrollment status
FR22_CODICE_ISTITUTO	chr	original-admin	> 10.000		School identification code
FR22_FL_CAMBIO_IST_AP		derived-admin	2	1 if inst21 $\neq$ inst22, 0 otherwise	Changed schools from one year to the next
FR22_FLAG_SCUOLA_DIG3	chr	derived-admin	2	substr(inst22,3,1)	Type: State or private
FR22_FLAG_SCUOLA_DIG34	chr	derived-admin	31	substr(inst22,3,2)	Type of institution
FR22_CARCERI	chr	derived-admin	2	substr(inst22,5,1)	Indicator for prison school
FR22_SERALE	chr	derived-admin	2	substr(inst22,8,1)	Indicator for evening school
VAR22	chr	original-admin	≈ 20		Level and year of attended school
FR22_FL_M3	chr	derived-admin	2	1 if VAR22=M-3, 0 otherwise	Final year of lower secondary school

Continued on next page...

Table A.1 Early estimation of school enrollment (IT) *(continued)*

Name	Type	Source	Dim.	Formula	Brief description
FR22_FL_BOC_AP	chr	derived-admin	2	1 if VAR21=VAR22, 0 otherwise	Individuals who have failed in the past
FR22_FL_INTPARIM3	chr	derived-admin	2	1 if VAR22=M-3 and private school	Private paritarian school indicator
FR22_PROCOM_SCU	chr	original-admin			Municipality of the attended school
FR22_FL_STESSO_COM	chr	derived-admin	2	1 if COM_RES=SCU_COM	Enrolled in school in same municipality

Table A.2 Early estimation of school enrollment (IT) extra information

Name	Dataset 1 (pr./lower sec.)	Regressors in logistic model	Regressors in RF	Dataset 2 (upper sec.)	Regressors in RF
ETA22	x		x	x	x
SESSO	x	x	x	x	x
COD_CITTADINANZA	x			x	
FL_ITA	x	x		x	
AREA_CITTADINANZA	x		x (ordinal)	x	x (ordinal)
REG_RES	x		x (ord/dummy)	x	x (dummy)
PRO_RES	x	x (strata)	x (ordinal)	x	x (ordinal)
PROCOM_RES	x			x	
Time dependent variables - training set (school year 2020/2021)					
FL_GT14_21	x	x			
FR21_STATO_FREQUENZA	x	x (dummy)	x (dummy)	x	x (dummy)
FR21_CODICE_ISTITUTO	x			x	
FR21_FL_CAMBIO_IST_AP (fl_ch)	x	x	x	x	x
FR21_FLAG_SCUOLA_DIGIT3	x	x	x		
FR21_FLAG_SCUOLA_DIGIT34				x	x (dummy)
FR21_CARCERI	x	x	x	x	x
FR21_SERALE	x	x	x	x	x
VAR21	x		x (ordinal)	x	x (ordinal)
FR21_FL_M3	x	x			
FR21_FL_BOC_AP (fl_fail)	x			x	
FR21_FL_INTPARIM3	x	x			
FR21_PROCOM_SCU	x			x	
FR21_FL_STESSO_COM (fl_mun)	x	x	x	x	x
Time dependent variables - test set (school year 2021/2022)					
FL_GT14_22	x	x			
FR22_STATO_FREQUENZA	x	x (dummy)	x (dummy)	x	x (dummy)
FR22_CODICE_ISTITUTO	x			x	
FR22_FL_CAMBIO_IST_AP (fl_ch)	x	x	x	x	x
FR22_FLAG_SCUOLA_DIGIT3	x	x	x		
FR22_FLAG_SCUOLA_DIGIT34				x	x (dummy)
FR22_CARCERI	x	x	x	x	x
FR22_SERALE	x	x	x	x	x
VAR22	x		x	x	x

Continued on next page...

Table A.2 Early estimation of school enrollment (IT) extra info *(continued)*

Name	Dataset 1	Log. Model	RF	Dataset 2	RF
FR22_FL_M3	x	x			
FR22_FL_BOC_AP (fl_fail)	x	x		x	
FR22_FL_INTPARIM3	x	x			
FR22_PROCOM_SCU	x			x	
FR22_FL_STESSO_COM (fl_mun)	x	x	x	x	x

A.2. Early imputation in the industrial turnover index in Germany (DE)

Table A.3 Early estimation of industrial turnover index in Germany (DE)

Name	Type	Source	Dimension	Formula	Brief description
BJ, BM	num	original	time		Reference year and month.
EF6U1	char	original	geog.		Land (Region).
EF4	char	original			Type of unit (single or multi-unit enterprise).
lag01_EF16_sc	num	derived	time	$lag01\_EF16 := shift(EF16)$ by $EF15ID$; $lag01\_EF16\_sc := scale(lag01\_EF16)$	Previous month's number of employed persons.
quantile_MA12_EF18_NACEdiv_1, quantile_MA12_EF19_NACEdiv_1	num	derived	time	$MA12\_EFk := rollmean(lag01\_EFk, 12)$; quant $:= ecdf(MA12\_EFk)$ by $WZ2$	Moving 12-month averages of turnover divided into groups within the economic sector (WZ) 2-digit codes.
lag01_EF18_sc, lag01_EF19_sc	num	derived	time	$lag01\_EFk := shift(EFk)$ by $EF15ID$; $lag01\_EFk\_sc := scale(lag01\_EFk)$	Previous month's value for turnover (domestic and foreign).
lag12_EF18_sc, lag12_EF19_sc	num	derived	time	$lag12\_EFk := shift(EFk, 12)$ by $EF15ID$; $lag12\_EFk\_sc := scale(lag12\_EFk)$	Previous year's value for turnover (domestic and foreign).
MA03_EF18_sc, MA03_EF19_sc	num	derived	time	$MA03\_EFk := rollmean(lag01\_EFk, 3)$; $MA03\_EFk\_sc := scale(MA03\_EFk)$	Moving averages turnover (3-month period), scaled between 0 and 1.
MA06_EF20_sc, MA06_EF21_sc	num	derived	time	$MA06\_EFk := rollmean(lag01\_EFk, 6)$; $MA06\_EFk\_sc := scale(MA06\_EFk)$	Moving averages of incoming orders (6-month period), scaled between 0 and 1.
lag01_PreisEF18_sc, lag01_PreisEF19_sc	num	derived	time	$Preis\_merged$ by $(Date, EF15)$; $lag01\_Preis\_sc := scale(shift(Preis))$	Previous month's prices based on WZ 4-digit, aligned to target variable.
kfact_EF18_sc, kfact_EF19_sc	num	derived	time	$kfactor\_merged$ by $(Date, EF15)$; $kfact\_sc := scale(kfactor)$	Calendar factors based on main WZ 4-digit.

A.3. Early imputation in the industrial turnover index in Spain (ES)

Table A.4 Early estimation of industrial turnover index in Spain (ES) regressors

Name	Type	Source	Dimension	Formula	Brief description
code_NUTS2_ent_ed	char	original	geog.		Edited value of the NUTS2 code for the enterprise.
code_NUTS2_val_1	char	original	geog.		Validated value of the NUTS2 code.
code_prov_ent_ed	char	original	geog.		Edited value of the code for the province.
code_prov_ed	char	original	geog.		Edited value of the code for the province.
code_munic_ent_ed	char	original	geog.		Edited municipality code.
code_postal_ent_ed	char	original	geog.		Edited postal code of the enterprise.
year_ref	char	original	time		Year of the reference time period.
month_ref	char	original	time		Month of the reference time period.
day_ref	char	original	time		Day of the reference time period.
nmonthsi_imputd_xprt	num	derived	time	$\sum_{j=1}^{i} \delta(z_k^{m-j} = \text{expert})$	Months manually imputed by an expert.
nmonthsi_imputd_auto	num	derived	time	$\sum_{j=1}^{i} \delta(z_k^{m-j} = \text{auto})$	Months automatically imputed.
nmonthsi_trnover0	num	derived	time	$\sum_{j=1}^{i} \delta(z_k^{(m-j)y} = 0)$	Months where target variable is 0.
nmonths13_notinsample	num	derived	time	$\sum_{j=1}^{13} \delta(k \notin U_c^{(m-j)y})$	Months unit is not in the sample.
code_NACE2class_frame_ed	char	original	econ. act.		Edited NACE Rev. 2 code (class) from frame.
code_NACE2class_frame_ent_ed	char	original	econ. act.		Edited NACE Rev. 2 code (enterprise).
code_NACE2class_ed	char	original	econ. act.		Edited NACE Rev. 2 code (data collection).
code_NACE2class_val_1	char	original	econ. act.		Validated NACE Rev. 2 code.
match_NACE2class_ed_val_1	num	derived	econ. act.	$\delta(\text{code\_ed}_k = \text{code\_val}_k)$	Binary indicating match in NACE class.
code_NACE2group_ed	char	original	econ. act.		Edited NACE Rev. 2 group.
code_NACE2group_ent_val_1	char	original	econ. act.		Validated NACE Rev. 2 group (enterprise).
match_NACE2group_est_ent_1	num	derived	econ. act.	$\delta(\text{group\_ed}_k = \text{group\_val}_k)$	Binary indicating match in NACE group.
code_NACE2div_ed	char	original	econ. act.		Edited NACE Rev. 2 division.
code_NACE2div_ent_val_1	char	original	econ. act.		Validated NACE Rev. 2 division.

Continued on next page...

Table A.4 Early estimation of industrial turnover index in Spain (ES) *(continued)*

Name	Type	Source	Dimension	Formula	Brief description
match_NACE2div_est_ent_1	num	derived	econ. act.	$\delta(\text{div\_ed}_k = \text{div\_val}_k)$	Binary indicating match in NACE division.
code_NACE2sect_ed	char	original	econ. act.		Edited NACE Rev. 2 section.
code_NACE2sect_ent_val_1	char	original	econ. act.		Validated NACE Rev. 2 section.
match_NACE2sect_est_ent_1	num	derived	econ. act.	$\delta(\text{sect\_ed}_k = \text{sect\_val}_k)$	Binary indicating match in NACE section.
code_NACE2MIG_ed	char	original	econ. act.		Edited code for Eurostat MIGS.
code_NACE2MIG_ent_val_1	char	original	econ. act.		Validated code for Eurostat MIGS.
trnovr_val_i	num	original	target		Validated turnover value for month m-i.
MAi_trnovr_val	num	derived	target	$\frac{1}{i}\sum_{j=1}^{i} z_k^{(m-j)y,\text{val}}$	Moving average of validated turnover.
q95_MAitrnovr_val_NACE2div	num	derived	target	$Q_{0.95}^{\text{NACE2div}}(\text{MAi}(z_k))$	Quantiles 0.95 across NACE division.
q95_MAitrnovr_val_NUTS2NACE2divEnt	num	derived	target	$Q_{0.95}^{\text{NUTS2Ent}}(\text{MAi}(z_k))$	Quantiles 0.95 across NUTS2 and Enterprise.
q95_MAitrnovr_val_NUTS2NACE2div	num	derived	target	$Q_{0.95}^{\text{NUTS2Div}}(\text{MAi}(z_k))$	Quantiles 0.95 across NUTS2 and Division.
above_q95_MAitrnovr_val_NACE2div	num	derived	target	$\delta(\text{MAi} \geq Q_{0.95})$	Binary for values above q95 (Division).
above_q95_MAitrnovr_val_NUTS2NACE2divEnt	num	derived	target	$\delta(\text{MAi} \geq Q_{0.95})$	Binary for values above q95 (NUTS2/Ent).
above_q95_MAitrnovr_val_NUTS2NACE2div	num	derived	target	$\delta(\text{MAi} \geq Q_{0.95})$	Binary for values above q95 (NUTS2/Div).
prob_trnovr_val_i_MA12_NACE2div	num	derived	target	$F^*_{\text{MA12}}(z_k^{(m-i)y,\text{val}})$	Empirical cumulative distribution (Division).
prob_trnovr_val_i_MA12_NUTS2NACE2divEnt	num	derived	target	$F^*_{\text{MA12\_Ent}}(z_k^{(m-i)y,\text{val}})$	Empirical cumulative distribution (Ent).
prob_trnovr_val_i_MA12_NUTS2NACE2div	num	derived	target	$F^*_{\text{MA12\_NUTS2}}(z_k^{(m-i)y,\text{val}})$	Empirical cumulative dist. (NUTS2/Div).
cv_sditrnovr_val_MAitrnovr_val	num	derived	target	$\frac{\sqrt{\frac{1}{i-1}\sum(z_k-\bar{z})^2}}{\bar{z}}$	Coefficient of variation of turnover.
min_trnovr_val_i	num	derived	target	$\min(z_k^{(m-1)y}, \ldots, z_k^{(m-i)y})$	Minimum turnover in last i months.
max_trnovr_val_i	num	derived	target	$\max(z_k^{(m-1)y}, \ldots, z_k^{(m-i)y})$	Maximum turnover in last i months.

Continued on next page...

Table A.4 Early estimation of industrial turnover index in Spain (ES) *(continued)*

Name	Type	Source	Dimension	Formula	Brief description
mean_trnovr_ed_NACE2group	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NACE group.
mean_trnovr_ed_NUTS2NACE2group	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NUTS2/Group.
mean_trnovr_ed_NACE2div	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NACE division.
mean_trnovr_ed_NUTS2NACE2div	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NUTS2/Div.
mean_trnovr_ed_NACE2section	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NACE section.
mean_trnovr_ed_NUTS2NACE2section	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NUTS2/Section.
mean_trnovr_ed_NACE2class	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NACE class.
mean_trnovr_ed_NUTS2NACE2class	num	derived	target	$\frac{1}{N}\sum z_k^{my,\text{ed}}$	Mean of edited turnover by NUTS2/Class.
sd_trnovr_ed_NACE2group	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NACE group.
sd_trnovr_ed_NUTS2NACE2group	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NUTS2/Group.
sd_trnovr_ed_NACE2div	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NACE division.
sd_trnovr_ed_NUTS2NACE2div	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NUTS2/Div.
sd_trnovr_ed_NACE2section	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NACE section.
sd_trnovr_ed_NUTS2NACE2section	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NUTS2/Section.
sd_trnovr_ed_NACE2class	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NACE class.
sd_trnovr_ed_NUTS2NACE2class	num	derived	target	$\sqrt{\frac{\sum(z_k-\bar{z})^2}{N-1}}$	Std deviation by NUTS2/Class.
rate_trnovr_ed0_vali	num	derived	target	$\frac{z_k^{my}-z_k^{(m-i)y}}{z_k^{(m-i)y}}$	Relative variation rate (Edited vs Val).
rate_trnovr_val1_vali	num	derived	target	$\frac{z_k^{m-1}-z_k^{m-i}}{z_k^{m-i}}$	Relative variation rate (Val vs Val).
rate_meanTrnovr_ed0_vali_NUTS2NACE2div	num	derived	target	$\frac{\bar{z}^{\text{ed}}_{my}-\bar{z}^{\text{val}}_{m-i}}{\bar{z}^{\text{val}}_{m-i}}$	Relative variation of mean turnover (NUTS2/Div).
rate_meanTrnovr_val1_vali_NUTS2NACE2div	num	derived	target	$\frac{\bar{z}^{m-1}-\bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (Validated).

Continued on next page...

Table A.4 Early estimation of industrial turnover index in Spain (ES) *(continued)*

Name	Type	Source	Dimension	Formula	Brief description
rate_meanTrnovr_ed0_vali_NACE2div	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (Division).
rate_meanTrnovr_val1_vali_NACE2div	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (Division Validated).
rate_meanTrnovr_ed0_vali_NACE2group	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (Group).
rate_meanTrnovr_val1_vali_NACE2group	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (Group Validated).
rate_meanTrnovr_ed0_vali_NUTS2NACE2group	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (NUTS2/Group).
rate_meanTrnovr_val1_vali_NUTS2NACE2group	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (NUTS2/Group Val).
rate_meanTrnovr_ed0_vali_NUTS2NACE2divEnt	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (NUTS2/Div Ent).
rate_meanTrnovr_val1_vali_NUTS2NACE2divEnt	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (NUTS2/Div Ent Val).
rate_meanTrnovr_ed0_vali_NUTS2NACE2class	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (NUTS2/Class).
rate_meanTrnovr_val1_vali_NUTS2NACE2class	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (NUTS2/Class Val).
rate_meanTrnovr_ed0_vali_NACE2class	num	derived	target	$\frac{\bar{z}^{\text{ed}} - \bar{z}^{\text{val}}}{\bar{z}^{\text{val}}}$	Mean variation rate (Class).
rate_meanTrnovr_val1_vali_NACE2class	num	derived	target	$\frac{\bar{z}^{m-1} - \bar{z}^{m-i}}{\bar{z}^{m-i}}$	Mean variation rate (Class Val).
IPI_ed_0_NUTS2NACE2class	num	derived	external		Industrial Production Index (Edited).
IPI_ed_0_NACE2class	num	derived	external		IPI by NACE Class.
IPI_ed_0_NUTS2NACE2group	num	derived	external		IPI by NUTS2/Group.
IPI_ed_0_NACE2group	num	derived	external		IPI by NACE Group.
rate_IPI_ed0_val1_NUTS2NACE2class	num	derived	external	$\frac{\text{IPI}^{\text{ed}} - \text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Variation rate of IPI (NUTS2/Class).
rate_IPI_ed0_val1_NACE2class	num	derived	external	$\frac{\text{IPI}^{\text{ed}} - \text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Variation rate of IPI (Class).
rate_IPI_ed0_val12_NUTS2NACE2class	num	derived	external	$\frac{\text{IPI}^{my,\text{ed}} - \text{IPI}^{m(y-1),\text{val}}}{\text{IPI}^{m(y-1),\text{val}}}$	Yearly variation rate of IPI.
rate_IPI_ed0_val12_NACE2class	num	derived	external	$\frac{\text{IPI}^{\text{ed}} - \text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Yearly variation rate of IPI (Class).
rate_yearToDateMeanIPI_ed0_val12_NUTS2NACE2class	num	derived	external	$\frac{\text{y2dMean}(\text{IPI}^{\text{ed}}) - \text{y2dMean}}{\text{y2dMean}}$	Y2D mean variation rate of IPI.
rate_yearToDateMeanIPI_ed0_val12_NACE2class	num	derived	external	$\frac{\text{y2dMean}(\text{IPI}^{\text{ed}}) - \text{y2dMean}}{\text{y2dMean}}$	Y2D mean variation rate (Class).

Continued on next page...

Table A.4 Early estimation of industrial turnover index in Spain (ES) *(continued)*

Name	Type	Source	Dimension	Formula	Brief description
rate_MA12IPI_ed0_val12_NUTS2NACE2class	num	derived	external	$\frac{\text{MA12(IPI}^{\text{ed}})-\text{MA12}}{\text{MA12}}$	12-month MA variation rate of IPI.
rate_MA12IPI_ed0_val12_NACE2class	num	derived	external	$\frac{\text{MA12(IPI}^{\text{ed}})-\text{MA12}}{\text{MA12}}$	12-month MA variation (Class).
rate_IPI_ed0_val1_NUTS2NACE2group	num	derived	external	$\frac{\text{IPI}^{\text{ed}}-\text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Variation rate of IPI (NUTS2/Group).
rate_IPI_ed0_val1_NACE2group	num	derived	external	$\frac{\text{IPI}^{\text{ed}}-\text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Variation rate of IPI (Group).
rate_IPI_ed0_val12_NUTS2NACE2group	num	derived	external	$\frac{\text{IPI}^{my,\text{ed}}-\text{IPI}^{m(y-1),\text{val}}}{\text{IPI}^{m(y-1),\text{val}}}$	Yearly variation rate (Group).
rate_IPI_ed0_val12_NACE2group	num	derived	external	$\frac{\text{IPI}^{\text{ed}}-\text{IPI}^{\text{val}}}{\text{IPI}^{\text{val}}}$	Yearly variation rate (Group).
rate_yearToDateMeanIPI_ed0_val12_NUTS2NACE2group	num	derived	external	$\frac{\text{y2dMean}^{\text{ed}}-\text{y2dMean}^{\text{val}}}{\text{y2dMean}^{\text{val}}}$	Y2D variation rate (NUTS2/Group).
rate_yearToDateMeanIPI_ed0_val12_NACE2group	num	derived	external	$\frac{\text{y2dMean}^{\text{ed}}-\text{y2dMean}^{\text{val}}}{\text{y2dMean}^{\text{val}}}$	Y2D variation rate (Group).
rate_MA12IPI_ed0_val12_NUTS2NACE2group	num	derived	external	$\frac{\text{MA12}^{\text{ed}}-\text{MA12}^{\text{val}}}{\text{MA12}^{\text{val}}}$	MA variation rate (NUTS2/Group).
rate_MA12IPI_ed0_val12_NACE2group	num	derived	external	$\frac{\text{MA12}^{\text{ed}}-\text{MA12}^{\text{val}}}{\text{MA12}^{\text{val}}}$	MA variation rate (Group).
IPRI_val_0_NUTS2NACE2class	num	derived	external		Industrial Price Index (NUTS2/Class).
IPRI_val_0_NACE2class	num	derived	external		Industrial Price Index (Class).
IPRI_val_0_NUTS2NACE2group	num	derived	external		Industrial Price Index (NUTS2/Group).
IPRI_val_0_NACE2group	num	derived	external		Industrial Price Index (Group).

A.4. Early imputation in accommodation establishments (PL)

Table A.5 Early estimation of accommodation establishments (PL) regressors

Name	Type	Source	Dimension	Formula	Brief description
X	int	technical	identifier		Technical row identifier
regon	char	original	identifier		Identifier of the accommodation object (REGON code)
voivodeship	int	original	geog.		Code of the voivodeship
county	int	original	geog.		Code of the county
object_type	int	original	categ.		Type/category of the accommodation object
municipality	int	original	geog.		Municipality (gmina) code
is_seasonal	int	original	categ.		Indicates if the object is seasonal (1/0)
period_yyyymm	int	original	temporal	YYYYMM	Year–month identifier
tourists	num	original	numerical		Number of tourists in the given month
month_num	int	derived	temporal		Month number (1–12)
active_in_month	int	derived	categ.		Indicates if object was active
seasonality	int	derived	categ.		Seasonal classification of the object
year	int	derived	temporal		Year extracted from period_yyyymm
month	int	derived	temporal		Month extracted from period_yyyymm
tourists_minus1	num	derived	lag	lag(tourists,1)	Tourists one month earlier
tourists_minus2	num	derived	lag	lag(tourists,2)	Tourists two months earlier
tourists_minus3	num	derived	lag	lag(tourists,3)	Tourists three months earlier
Pandemia	num	derived	policy		Pandemic restriction intensity (0–1)
time_trend	int	derived	temporal	$(y - min(y))12 + m$	Linear time trend (month index)
quarter	int	derived	temporal	quarter(m)	Calendar quarter (1–4)
winter	int	derived	seasonal	$m \in \{12, 1, 2\}$	Winter dummy
spring	int	derived	seasonal	$m \in \{3, 4, 5\}$	Spring dummy
summer	int	derived	seasonal	$m \in \{6, 7, 8\}$	Summer dummy
autumn	int	derived	seasonal	$m \in \{9, 10, 11\}$	Autumn dummy
lockdown	int	derived	policy	$Pandemia \geq th$	Binary lockdown indicator
Lockdown_owid	num	external	policy	from OWID	External COVID restriction index
avg_voivodeship_object_type	num	derived	comp.	mean(tourists)	Avg tourists per object type in voivodeship
mean_win_3_by_regon_object_type	num	derived	rolling	rollmean(t, 3)	3-month rolling mean by object type
mean_win_6_by_regon_object_type	num	derived	rolling	rollmean(t, 6)	6-month rolling mean by object type
mean_win_12_by_regon_object_type	num	derived	rolling	rollmean(t, 12)	12-month rolling mean by object type
median_win_6_by_object_type	num	derived	rolling	rollmedian(t, 6)	6-month rolling median
q25_win_3_by_object_type	num	derived	rolling	$Q_{0.25}(t, 3)$	25th percentile over 3 months
q75_win_12_by_object_type	num	derived	rolling	$Q_{0.75}(t, 12)$	75th percentile over 12 months
mean_win_3_by_voivodeship_county_object_type	num	derived	rolling	rollmean(t, 3)	3-month rolling mean by location and type
median_win_12_by_voivodeship_county_object_type	num	derived	rolling	rollmedian(t, 12)	12-month rolling median by location/type

Continued on next page...

Table A.5 Early estimation of accommodation establishments (PL) *(continued)*

Name	Type	Source	Dimension	Formula	Brief description
q25_win_6_by_voivodeship_county_object_type	num	derived	rolling	$Q_{0.25}(t, 6)$	25th percentile by location/type
sum_win_12_by_regon_voivodeship_county	num	derived	roll-sum	rollsum(t, 12)	12-month rolling sum per location
share_3mo_vs_county	num	derived	share	$t/sum(county)$	Share of tourists in county (3 months)
share_12mo_vs_voivodeship	num	derived	share	$t/sum(voiv)$	Share in voivodeship (12 months)

Appendix B

Appendix: extended results

B.1. Early imputation in school enrollment (IT)

Table B.1 Precision of dataset1

PRECISION	**mtry**			
minLeafSplit	8	10	15	20
1	**25,92%**	**25,92%**	**25,85%**	**25,88%**
5	25,17%	25,44%	25,61%	25,75%
10	24,84%	25,20%	25,47%	25,60%
20	24,37%	24,84%	25,30%	25,51%

Table B.2 Precision of dataset2

PRECISION	**mtry**				
minLeafSplit	8	10	15	20	25
1	**55,91%**	**56,15%**	**56,10%**	**55,99%**	**55,93%**
5	55,45%	55,74%	55,98%	55,96%	55,89%
10	55,25%	55,63%	55,88%	55,87%	55,85%
20	54,96%	55,43%	55,81%	55,85%	55,85%

Table B.3 Recall of dataset1

RECALL	**mtry**			
minLeafSplit	8	10	15	20
1	**26,52%**	**26,54%**	**26,58%**	**26,66%**
5	25,91%	26,17%	26,32%	26,46%
10	25,59%	25,95%	26,20%	26,30%
20	25,15%	25,54%	25,98%	26,16%

Table B.4 Recall of dataset2

RECALL minLeafSplit	mtry 8	10	15	20	25
1	**54,68%**	**54,87%**	**55,77%**	**56,00%**	**56,14%**
5	54,11%	54,68%	55,46%	55,82%	55,99%
10	53,80%	54,45%	55,30%	55,64%	55,87%
20	53,35%	54,06%	54,96%	55,43%	55,75%

Table B.5 F1 score of dataset1

F1 SCORE minLeafSplit	mtry 8	10	15	20
1	**26,218%**	**26,228%**	**26,209%**	**26,263%**
5	25,535%	25,800%	25,961%	26,100%
10	25,213%	25,569%	25,830%	25,943%
20	24,754%	25,184%	25,634%	25,830%

Table B.6 F1 score of dataset2

F1 SCORE minLeafSplit	mtry 8	10	15	20	25
1	**55,285%**	**55,501%**	**55,933%**	**55,997%**	**56,035%**
5	54,769%	55,209%	55,719%	55,891%	55,941%
10	54,517%	55,035%	55,588%	55,758%	55,857%
20	54,144%	54,737%	55,384%	55,641%	55,801%

Table B.7 Absolute difference between percentage in dataset1

Abs.Diff minLeafSplit	mtry 8	10	15	20
1	**0,0283%**	**0,029%**	**0,035%**	**0,037%**
5	0,036%	0,035%	0,034%	0,034%
10	0,037%	0,036%	0,035%	0,034%
20	0,039%	0,035%	0,033%	0,031%

Table B.8 Absolute difference between percetage in dataset2

Abs.Diff minLeafSplit	mtry 8	10	15	20	25
1	**-0,287%**	**-0,299%**	**-0,078%**	**0,0018%**	**0,049%**
5	-0,317%	-0,248%	-0,122%	-0,032%	0,025%
10	-0,342%	-0,275%	-0,137%	-0,054%	0,005%
20	-0,382%	-0,324%	-0,199%	-0,097%	-0,022%

Table B.9 Relative difference between percentage in dataset1

Rel.Diff	**mtry**			
minLeafSplit	8	10	15	20
1	**2,297%**	**2,397%**	**2,850%**	**3,039%**
5	2,967%	2,879%	2,795%	2,734%
10	3,006%	2,958%	2,869%	2,739%
20	3,176%	2,821%	2,687%	2,540%

Table B.10 Relative difference between percentage in dataset2

Rel.Dif	**mtry**				
minLeafSplit	8	10	15	20	25
1	**2,199%**	**2,290%**	**0,596%**	**0,014%**	**0,372%**
5	2,425%	1,903%	0,931%	0,246%	0,192%
10	2,623%	2,108%	1,051%	0,415%	0,035%
20	2,922%	2,483%	1,525%	0,743%	0,167%

Table B.11 Runtime for dataset1

Runtime	**mtry**			
minLeafSplit	8	10	15	20
1	**19**	**22**	**30**	**38**
5	19	22	30	37
10	18	22	29	38
20	19	22	30	38

Table B.12 Runtime in dataset2

Runtime	**mtry**				
minLeafSplit	8	10	15	20	25
1	13	15	20	24	27
5	12	14	18	23	27
10	12	14	18	23	27
20	11	13	18	23	27

B.2. Early imputation in the industrial turnover index in Spain (ES)

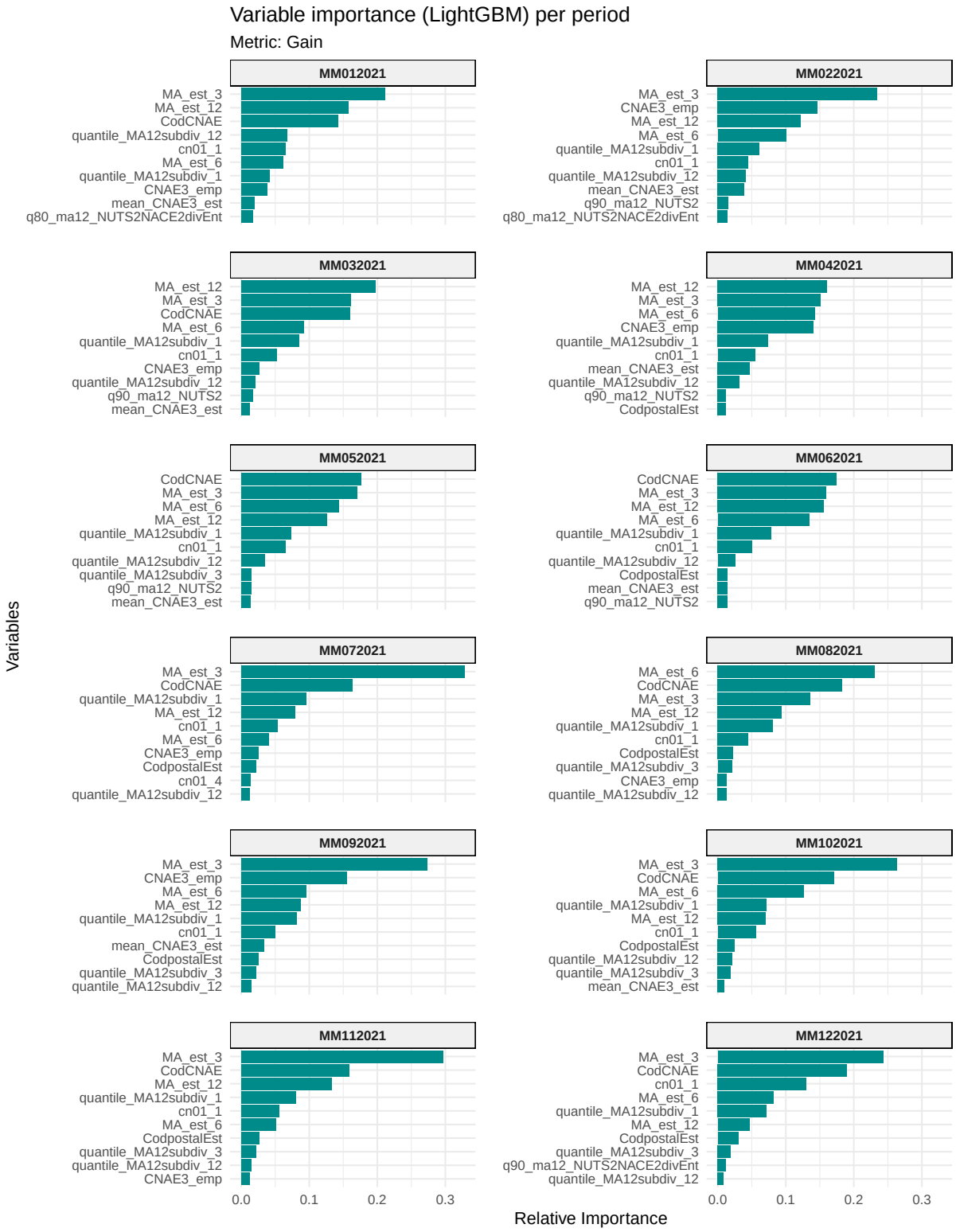

Figure B.1 Variable importance for periods from Jan, 2021 to Dec, 2021.

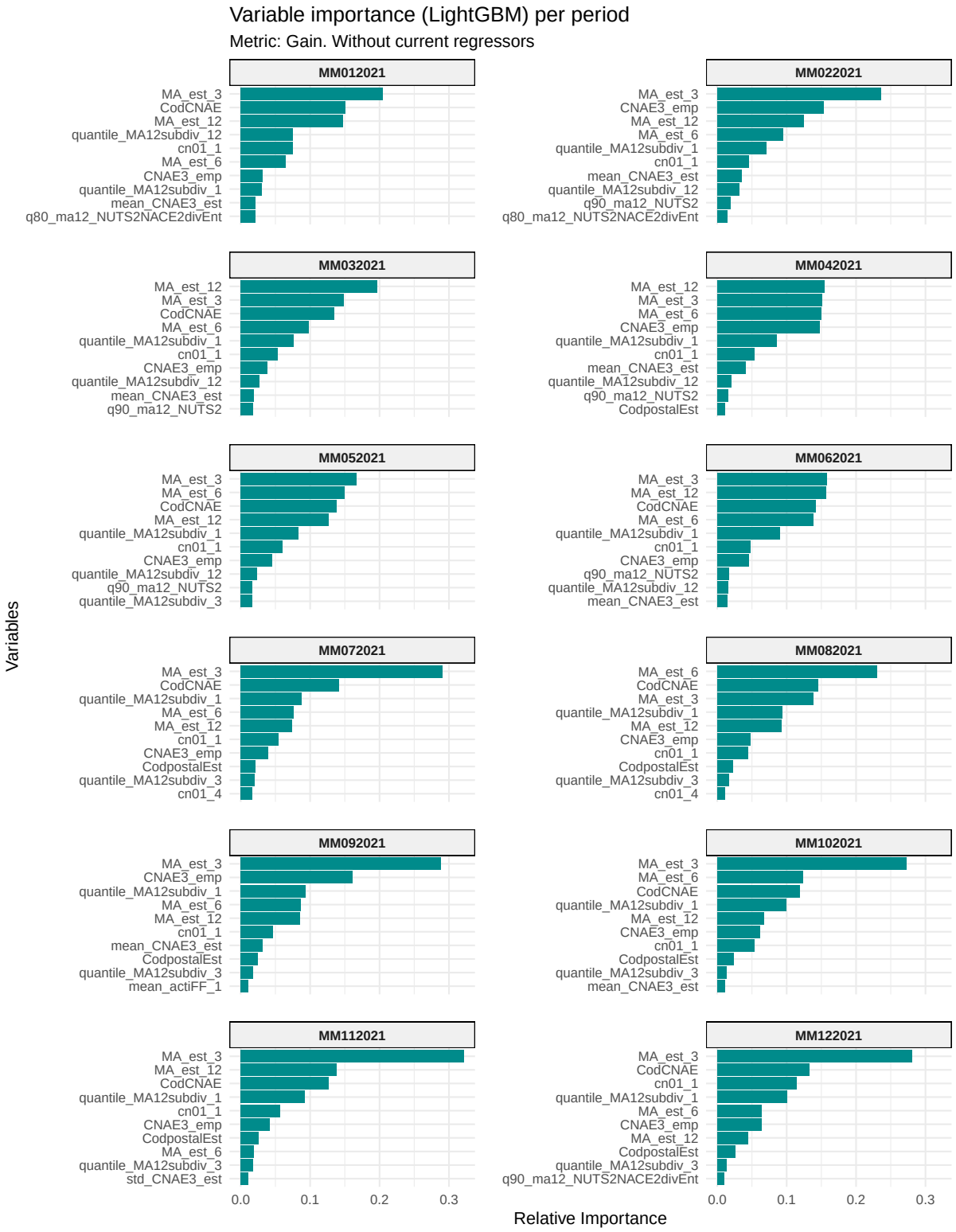

Figure B.2 Variable importance in the model without current regressors for periods from Jan, 2021 to Dec, 2021.

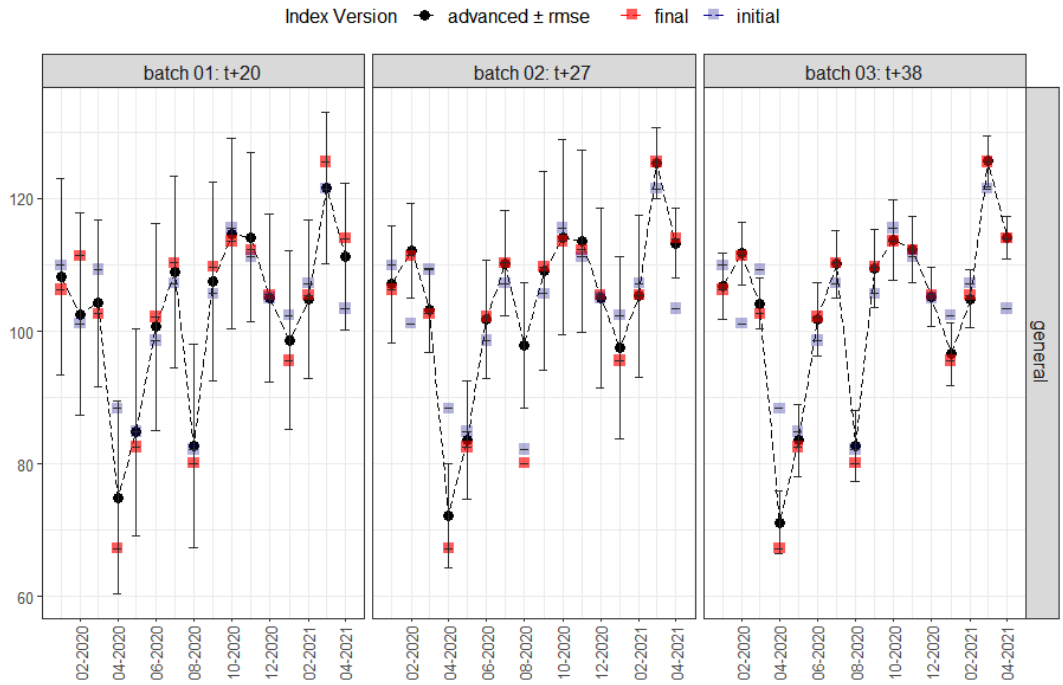

Figure B.3 General Index Series from Jan, 2020 to April, 2021 (executed in 2022).

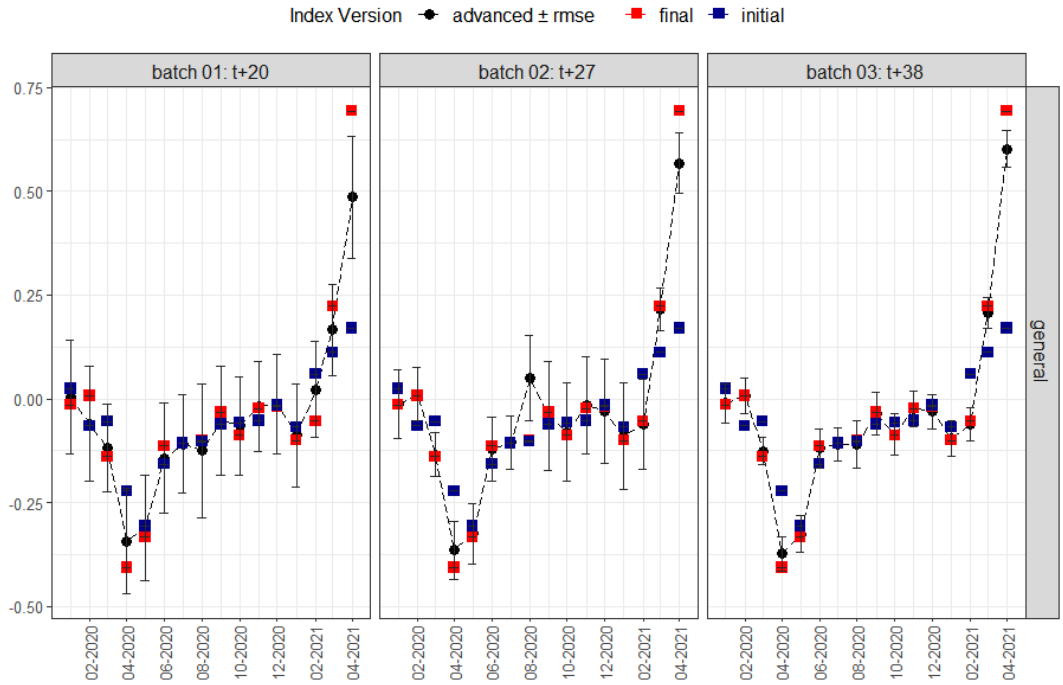

Figure B.4 Annual Variation Rates Series from Jan, 2020 to April, 2021 (executed in 2022).

Table B.13 Hyperparameters selected for the best model in each period

period	objective	metric	boosting	nrounds	eta	colsample_bytree	lambda_l1
MM012021	regression	mae	gbdt	300	0,01	0,8	1
MM022021	regression	mae	gbdt	300	0,01	0,8	1
MM032021	regression	mae	gbdt	300	0,01	0,8	1
MM042021	regression	mae	gbdt	300	0,01	0,8	1
MM052021	regression	mae	gbdt	300	0,01	0,8	1
MM062021	regression	mae	gbdt	300	0,01	0,8	1
MM072021	regression	mae	gbdt	300	0,01	0,8	1
MM082021	regression	mae	gbdt	300	0,01	0,8	1
MM092021	regression	mae	gbdt	300	0,05	0,8	1
MM102021	regression	mae	gbdt	300	0,01	0,8	1
MM112021	regression	mae	gbdt	300	0,05	0,8	1
MM122021	regression	mae	gbdt	300	0,05	0,8	1

Table B.14 Number of units per period and batch in total and to be predicted

period	**batch**	**size**	**n_pred**	**p_pred**
MM012021	20	11852	3668	30.95
MM012021	29	11852	1419	11.97
MM012021	38	11852	772	6.51
MM012021	FF	11557	0	0.00
MM022021	20	11515	3496	30.36
MM022021	29	11515	1256	10.91
MM022021	38	11515	880	7.64
MM022021	FF	11793	0	0.00
MM032021	20	11782	4175	35.44
MM032021	29	11782	1088	9.23
MM032021	38	11782	760	6.45
MM032021	FF	11814	0	0.00
MM042021	20	11742	3462	29.48
MM042021	29	11742	1180	10.05
MM042021	38	11742	916	7.80
MM042021	FF	11750	0	0.00
MM052021	20	11585	3433	29.63
MM052021	29	11585	1267	10.94
MM052021	38	11585	841	7.26
MM052021	FF	11588	0	0.00
MM062021	20	11574	3767	32.55
MM062021	29	11574	1420	12.27

Continued on next page...

Table B.14 Number of units per period and batch — (continued)

period	batch	size	n_pred	p_pred
MM062021	38	11574	1090	9.42
MM062021	FF	11580	0	0.00
MM072021	20	11190	5119	45.75
MM072021	29	11190	3387	30.27
MM072021	38	11190	1781	15.92
MM072021	FF	11206	0	0.00
MM082021	20	11199	4183	37.35
MM082021	29	11199	1851	16.53
MM082021	38	11199	1493	13.33
MM082021	FF	11198	0	0.00
MM092021	20	11197	4162	37.17
MM092021	29	11197	1571	14.03
MM092021	38	11197	1329	11.87
MM092021	FF	11235	0	0.00
MM102021	20	11215	3587	31.98
MM102021	29	11215	1690	15.07
MM102021	38	11215	1324	11.81
MM102021	FF	11217	0	0.00
MM112021	20	11199	4387	39.17
MM112021	29	11199	1847	16.49
MM112021	38	11199	1603	14.31
MM112021	FF	11211	0	0.00
MM122021	20	11200	4849	43.29
MM122021	29	11200	2010	17.95
MM122021	38	11200	1559	13.92
MM122021	FF	11216	0	0.00