UTRECHT UNIVERSITY

Department of Information and Computing Science

Applied Data Science master thesis

Predicting Fertility in the Netherlands: A Data-Driven Approach

First examiner:
Paulina Pankowska

Candidate:
Vincent Haverhoek

Second examiner:
Vincent Buskens

July 1, 2024

Abstract

Forecasting fertility trends is crucial for understanding demographic shifts and their societal implications. However, accurately predicting fertility patterns remains a challenge due to the complex interplay of economic, social, and individual factors. This study, part of the PreFer Data challenge, proposes a data-driven framework to predict fertility trends in the Netherlands. Leveraging the large-scale longitudinal LISS dataset, this research explores the dataset's suitability for predicting fertility by identifying key attributes related to demographic information, household characteristics, income, employment, and health metrics. Multiple models, including neural network, random forest, and linear regression classifiers, were trained and evaluated. The methodology involved initial stratified k-Fold Cross-Validation to find ideal hyperparameter combination, followed by bootstrap resampling to assess the impact of training data size and model robustness. The results demonstrate that AI-driven methods can effectively capture the underlying patterns in fertility data, with models achieving an average F1 score of 0.7 and showing strong 95% confidence interval (CI) values within 0.01, indicating reliable and consistent performance. This provides insights into the predictive potential of the LISS dataset. However, further research is needed to validate these findings across different data sources and incorporate additional relevant attributes to enhance predictive accuracy.

Contents

1. Introduction

Understanding and predicting fertility trends is key to grasping demographic changes and their impact on societies. By accurately forecasting these trends, we gain insights into future shifts in population dynamics, allowing for better preparation and adaptation to forthcoming changes. These trends shape the age distribution of a population, influencing everything from workforce dynamics to economic potential. A youthful, working-age majority can drive economic growth through a "demographic dividend" enhancing productivity and prosperity. On the other hand, a higher proportion of dependents, whether young or elderly, can strain the economy and challenge those in the workforce.[1].

Forecasting when and which people might be getting children is not all about predicting economic gains, either. Forecasting these trends can inform policies related to education, employment, and healthcare. For example, countries with declining fertility rates may need to adjust their policies to address potential labor shortages and increased demand for elderly care. Conversely, countries with high fertility rates may focus on expanding education and job opportunities to harness the potential of a growing young population[1]. By anticipating these demographic shifts, we can create societies that are better prepared and more adaptable, where resources are managed efficiently, and where people of all ages can benefit.

1.1 Postponement

No society or time period is the same, making forecasting a continuous challenge. In the case of fertility specifically, advanced societies have experienced a significant postponement transition over the past decades, characterized by a delay in childbearing age leading to low fertility rates [2]. This phenomenon has sparked extensive research interest for several reasons. Scholars aim to understand the underlying causes of this shift, which include changes in societal norms, economic conditions, and individual life choices. By examining these factors, researchers seek to explain how and why this postponement of fertility behaviours has evolved over time. [3].

The study of postponement transitions has made significant scientific contributions to the field of demography and related disciplines. Researchers have developed sophisticated models to analyze the tempo and quantum effects of fertility postponement [4]. These models have enhanced our understanding of how delayed childbearing affects period fertility rates and cohort completed fertility. Additionally, scholars have explored the biological and social mechanisms underlying the postponement phenomenon, shedding light on the complex interplay between individual decisions and societal trends [5]. This body of research has advanced theoretical frameworks and improved methodological approaches for studying fertility patterns in societies.

Scholars are also interested in the postponement transition to devise effective policy responses. By understanding the factors leading to delayed childbearing, policymakers can design interventions that support family planning and address potential negative consequences of low fertility, such as population aging and labor shortages. In this way, the research on postponement transitions not only contributes to academic knowledge but also informs practical solutions for societal challenges.

Understanding the postponement of childbearing is crucial because it affects population dynamics and has long-term implications for economic and social policies. For example, delayed childbearing can lead to lower overall fertility rates, impacting population growth and age structure. This, in turn, influences the planning and sustainability of social security systems, healthcare, and labor markets. Some European formerly lowest-low-fertility countries are witnessing increases in fertility rates, particularly as the transitory effects of delayed childbearing diminish[6], this shows that the fertility landscape remains dynamic and multifaceted. In Europe, subtle differences in total fertility rates (TFRs) carry profound implications for the long-term trajectory of population decline, highlighting the complexity of fertility dynamics [7].

Predicting fertility accurately is more important now than ever, particularly in Western countries, due to the phenomenon of postponement. The relatively new nature of this phenomenon means that traditional methods of predicting fertility

may no longer be adequate. This necessitates rethinking how we predict fertility, as accurate predictions have strong implications for fertility rates and the planning of social and economic policies. Particularly in the Dutch context, it is crucial to understand and address the postponement transition for developing informed and effective policy responses.

1.2 Prediction efforts

Despite longstanding efforts to predict birth rates dating back to the post-World War II era [8], accurate forecasts remain elusive due to the intricate interplay of economic, social, and individual factors. Early attempts relied on simplistic extrapolations of past trends or adjusted models based on changes in demographic composition [8]. Subsequently, more recent refinements incorporated socio-economic considerations in explanatory modelling[9][10]. While these explanatory models have advanced our understanding by providing theoretical mechanisms, their predictive accuracy or power remains rather low[11][12]. This is partly due to challenges such as overfitting, where a model captures idiosyncrasies of the data that fail to generalize[11].

1.3 Goal of research

With the advent of artificial intelligence (AI) and applied data science, new data-driven opportunities asire to extend upon these theoretical models[12] and to address the complexities of predicting fertility. In this paper, as part of the PreFer data challenge [13], I propose a fully data-driven approach to predict fertility trends in the Netherlands. Leveraging the large-scale longitudinal LISS dataset, which encompasses a diverse array of variables on Dutch households. The goal of the research is to explore the suitability of the LISS dataset to predict fertility in the Netherlands, from a data-centric perspective. This is done by identifying and processing relevant attributes, then trying multiple prediction models such as neural network, random forest and linear regression Classifiers. Finally, guided by sensitivity analyses, the models are evaluated to propose a suitable and robust approach to predicting fertility. This can then be incorporated into the overall findings of the PreFer Data challenge to enhance the understanding of the predictive modelling of fertility.

By utilizing AI and data science techniques, this approach aims to explore methods that could provide new ways to predict fertility trends compared to traditional methods, by incorporating and examining the predictive potential of the LISS dataset. Accurate fertility forecasts are essential for planning and policy-making, particularly in areas such as social security, healthcare, and labor markets. Secondly, this research will help identify the most relevant socio-economic and demographic factors influencing fertility in the Netherlands, offering insights that can inform targeted interventions and policies. Additionally, these insights could contribute to theoretical advancements in understanding fertility dynamics. By uncovering new predictors that may not have been previously theorized, this research could lay the groundwork for developing new theories or refining existing ones.

Moreover, the innovative use of the LISS dataset and advanced modelling techniques can set a precedent for future fertility research, demonstrating the potential of AI-driven methods when applied to traditional social science data sources such as longitudinal surveys. This study, along with other research in the PreFer data challenge, contributes to the overall findings of the challenge. It enhances the understanding of predictive modelling of fertility and provides a framework that can be adapted and applied to other countries and datasets. This could potentially lead to global improvements in fertility prediction and policy planning.

2. LISS Dataset

The LISS (Longitudinal Internet Studies for the Social Sciences) panel is a high-quality online survey infrastructure managed by the non-profit research institute Centerdata and based on a traditional probability sample drawn from the Dutch population register by Statistics Netherlands. The representativeness of the LISS panel is comparable to traditional surveys that use probability sampling, with initial selection biases corrected through periodic refreshment samples[13].

The LISS panel began in 2007 with approximately 5,000 households, including 8,000 individuals aged 16 and older (about 6,000 aged 18–45). With an annual attrition rate of about 10%, new panel members are recruited every two years to maintain representativeness. By 2020, around 10,000 people aged 18–45 had been part of the panel at some point, with about 6,900 participating in at least one Core survey from 2007 to 2020.

2.1 Data sources

The LISS panel comprises two main sources of data[13]. The first source is the LISS Core Study, which is a longitudinal study conducted annually. This study encompasses a set of ten modules that cover a wide range of topics, including income, education, health, values, religion, personality, and fertility behavior (such as fertility intentions).

The second source is the Background Survey. This survey is completed by a household's contact person upon joining the panel and is updated monthly. It collects basic socio-demographic information about the household and all its members, even those who do not participate in the Core surveys.

2.2 Structure and Scope

The Core Study modules and their various waves are stored separately. In the context of PreFer, to measure fertility outcomes, a merged dataset from all Core

Study modules from 2007 to 2020 was constructed by the organizers of the PreFer data challenge, comprising over 30,000 variables. The dataset aims to predict who will have a child between 2021 and 2023 based on data from previous years. The target group consists of individuals who were between 18 and 45 years old in 2020 and participated in at least one Core study between 2007 and 2020[13].

2.3 Outcome variable

Despite most of the target group dropping out by 2021–2023, an outcome variable could be created for about 1,400 respondents, almost all of whom participated in at least one Core study in 2019–2020. The binary outcome variable indicates whether a respondent had a child between 2021 and 2023. For participants in the PreFer data challenge, a dataset containing 987 outcome variables is available, the remaining labels of the 1400 respondents are holdout and only available to the challenge organizers for further validation. In this dataset, approximately 25% of this group had a positive outcome (i.e., they had a child), while 75% had a negative outcome (no child). This sample size, while small, is typical for social science datasets with representative samples and provides a unique longitudinal dataset for research purposes.

2.4 Missing Data

Upon analyzing the PreFer dataset, which contains 31,635 attributes (excluding background data), several key statistics regarding missing data were identified.

2.4.1 Overall Missingness

The core dataset exhibits a high degree of missingness:

- Total Missing Values (All Data): approximately 180 million
- Percentage of Missing Values (All Data): 88.52%

A subset of the data, filtered based on the availability of the outcome variable (having a new child or not), resulted in 987 records. The missing data statistics for this subset are:

- Total Missing Values (Labelled Data): 24.3 million
- Percentage of Missing Values (Labelled Data): 77.82%

2.4.2 Columns with High Missingness

The proportion of columns with significant missing data was also examined:

2.4.2.1 Entire Dataset

- Columns with <10% Missing: 5 (0.015%)
- Columns with <20% Missing: 6 (0.019%)
- Columns with <30% Missing: 6 (0.019%)

2.4.2.2 Subset of Entire Dataset with available outcome labels

- Columns with <10% Missing: 275 (0.869%)
- Columns with <20% Missing: 788 (2.49%)
- Columns with <30% Missing: 1550 (4.90%)

These statistics underscore the significant extent of missing data in the core dataset. This high degree of missingness can be attributed to several factors, such as the design of aggregating surveys, where each column often represents an individual question from a specific wave of the survey, resulting in different columns for different waves of the same question. Consequently, there are no prevalent attributes (with more than 70% availability) in the full dataset. The few columns with less than 30% missing data are primarily IDs and label indicators, which are not particularly relevant for in-depth analysis.

In contrast, when examining the labelled subset, which consists of records suitable for prediction due to the presence of outcome labels, the availability of data within attributes improves significantly. Although the data remains quite sparse, there are many attributes that are available for a large portion of the labelled data, making this subset more viable for predictive analysis.

Overall, the analysis highlights the challenge posed by the extensive missing data and the sparseness in the core dataset used for PreFer, particularly in the context of its sparse variables. For the sake of the data challenge, this underscores the importance of focusing on the already widely available attributes in regarding to the entries that have an outcome variable, for any meaningful predictive modelling.

3. Method

In this chapter, I outline the methodological approach employed in this study to address the task of predicting whether an individual in the Netherlands will have a(nother) child or not within the next 3 years. This includes feature selection strategies and handling missing data, preprocessing steps, creation of training data, and model selection and training techniques. The methods chosen are designed to account for the class imbalance present in the dataset and to provide robust assessments of model performance.

3.1 Data Selection and imputation

As described in Section 2.4, the core dataset contains a substantial amount of missing and widely dispersed data. To address this, I select attributes with high availability (75%+ for labelled data rows) and impute missing values to make the data usable for prediction models. I then preprocess the data by normalizing and transforming the attributes, ensuring consistency. Finally, I use `RandomForestClassifier` as a feature selection method, to identify the best attributes for prediction, based on feature importance for predicting the outcome variable of having a new child or not.

3.1.1 Handling and imputing Missing Data

To manage missing data effectively, I first set a threshold for missingness at 75% within the section where outcomes are available. This threshold is based on the analysis in Section 2.4. The 75% threshold was chosen after careful consideration of data availability across different percentages. Any attribute with less than 75% value availability for labelled entries, as well as those containing indexes, IDs, or directly related to the outcome label, was excluded from further analysis, reducing the number of dimensions from 31,635 to 872.

For the remaining columns, I utilized the `IterativeImputer` from the `scikit-learn` library to impute missing values in numerical columns and the `SimpleImputer` to

impute missing values in categorical columns. Imputation was based on the entire dataset, including entries that do not have outcome labels. `IterativeImputer` operates by iteratively estimating missing values for each numerical feature using the observed values of other numerical features[14]. It models each feature with missing values as a function of other features and uses this model to predict missing values. This iterative process continues until convergence, effectively imputing missing data.

For the categorical attributes, I employed the `SimpleImputer` with the `most_frequent` strategy. This method fills in missing values with the most frequent (mode) value of each categorical column. This approach is relatively robust due to the high availability (greater than 75%) of data for each categorical attribute, ensuring that the imputed values accurately reflect the existing data.

After imputation, I filtered the dataset to include only those rows where the outcome variable (having a new child) is present, ensuring that the dataset used for further analysis is complete and appropriate for predictive modelling.

3.1.2 Preprocessing

After imputation, the dataset undergoes preprocessing to prepare it for modelling. Numerical variables are transformed using the `QuantileTransformer` from `scikit-learn`. This transformation maps the data to a Gaussian distribution with values ranging approximately between 0 and 1, reducing the impact of outliers and ensuring a more uniform scale across features [15]. This step is crucial for enhancing the robustness of machine learning models to variations in data distribution, particularly benefiting distance-based models.

Following the transformation of numerical variables, categorical variables are encoded using one-hot encoding. This technique converts categorical variables into binary vectors, where each category is represented as a binary feature[16]. This approach preserves the categorical nature of the variables while making them suitable for machine learning algorithms that expect numerical input.

One-hot encoding is typically applied before feature selection to ensure that all categorical levels are considered during the feature selection process. By representing

each category as a separate binary feature, one-hot encoding allows the machine learning algorithm to assess the importance of each individual category in relation to the target variable[16]. This approach was chosen to ensure comprehensive consideration of all categorical levels in the predictive modelling process.

In some cases, one-hot encoding might be applied after feature selection. This approach could be chosen if the initial dataset contains a large number of categorical variables, and feature selection aims to reduce dimensionality by focusing on the most relevant features regardless of their original form. Nevertheless, for this particular analysis, I opted to apply one-hot encoding before feature selection to ensure comprehensive consideration of all categorical levels in the predictive modelling process. This means that for categorical features, some category values may not be included after feature selection. This doesn't imply that those values are disregarded entirely; rather, it indicates that knowing the presence or absence of a strong predictor category is sufficient, regardless of the other specific categories represented.

3.1.3 Feature Selection

To identify the most relevant variables for predicting the outcome of having a new child or not, I employed a `RandomForestClassifier` from `scikit-learn` for feature selection with `n_estimators = 1000`. `RandomForestClassifier` measures feature importance by the reduction in impurity, specifically the Gini index, brought by each feature across all 1000 decision trees in the forest[17].

To determine the optimal number of features, I used the `feature_importances_` attribute[18] of `RandomForestClassifier` and manually set a Gini importance threshold of 0.0025. This threshold was chosen based on an analysis of feature importances plotted in Figure 3.1, to retain features that significantly contribute to predicting the target variable while excluding less informative ones.

Figure 3.1 shows the feature importances, with the chosen threshold of 0.0025 indicated by the red dashed line. This threshold represents a fine balance for feature selection. A higher threshold of 0.005 would exclude more features where the first significant dip starts and a lower threshold of 0.002 would include more features extending into a long tail. After testing the remaining part of the pipeline for these

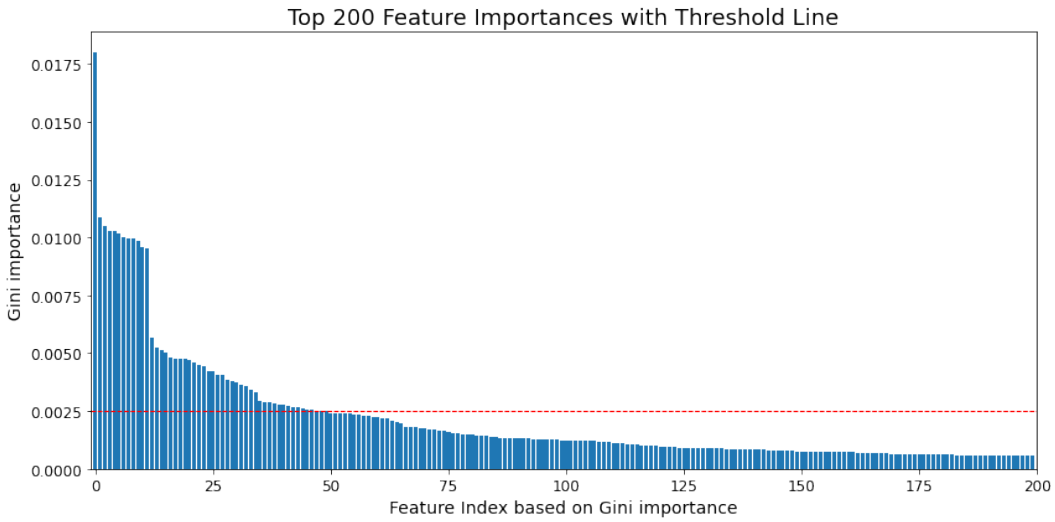

Figure 3.1: Top 200 Feature Importances with Threshold Line at 0.0025 Gini importance

three thresholds, the threshold of 0.0025 appears to be optimal in general, ensuring that only the most influential variables are retained, reducing overfitting and improving the model's generalization capability.

3.1.3.1 Introduction to Random Forest Classifier and Gini Importance

A Random Forest Classifier works by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes of the individual trees. Each tree is constructed using a random subset of features and samples, a technique known as bootstrap aggregating or bagging. This process helps in reducing the variance of the model and improving its robustness[17].

Feature selection in a Random Forest is based on the concept of Gini importance. The Gini index measures the impurity of a node, whereas a lower Gini index indicates a purer node. During the construction of each tree, splits are made to decrease this impurity. The Gini importance of a feature is computed as the total reduction of the Gini index brought by that feature, averaged over all trees in the forest[16]. Features that lead to greater reductions in impurity (higher Gini importance) are considered more important.

3.2 Creating Training Data

The final training dataset is a composite of attributes with high availability, selected based on their variation in relation to the outcome variable, and preprocessed to be imputed, normalized and usable for prediction models. When creating the training dataset to use on the prediction models, I focused on addressing the class imbalance issue present in the dataset. The outcome variable in the LISS dataset consists of 987 instances, with a distribution of approximately 1/4 positive (New child) outcomes and 3/4 negative (No new child) outcomes. There are many methods to combat this class imbalance and facilitate robust model training[19], I employ two distinct methods: stratified sampling and random over-sampling.

3.2.1 Stratified Test Split

Initially, I performed a stratified 80/20 train-test split on the dataset. This involves partitioning the data into a training set, which constitutes 80% (789) of the 987 total instances, and a test set, which comprises the remaining 20% (198). The stratified approach ensures that the distribution of positive and negative outcomes remains consistent across both the training and test sets, thus maintaining the integrity of the dataset's original distribution.

3.2.2 Training sets

After generating the stratified test split, two separate training sets will be employed to mitigate the class imbalance observed within the training data. Both sets will be utilized in training the models.

- **Stratified Training Set:** This approach utilizes the 80% stratified training set generated earlier, maintaining the original class proportions.
- **Random Over-sampling:** In addition to the stratified training set, I create an oversampled version where the positive outcomes are artificially increased to match the number of negative outcomes. Random over-sampling involves randomly duplicating instances from the minority class (positive outcomes) until the class distribution is balanced. This can provide the model with more instances of the minority class, potentially improving its ability to learn patterns and make accurate predictions for positive outcomes[19].

These two methods offer complementary approaches to address class imbalance,

each with its advantages. The stratified training set maintains the original data distribution, ensuring that the model is trained on a representative sample of the overall dataset. On the other hand, the random over-sampling technique artificially balances the classes by increasing the number of samples in the minority class. This approach can potentially enhance the model's performance on the minority class by providing more instances of rare outcomes for the model to learn from. The difference in class distribution between the two methods can be seen in Figure 3.2.

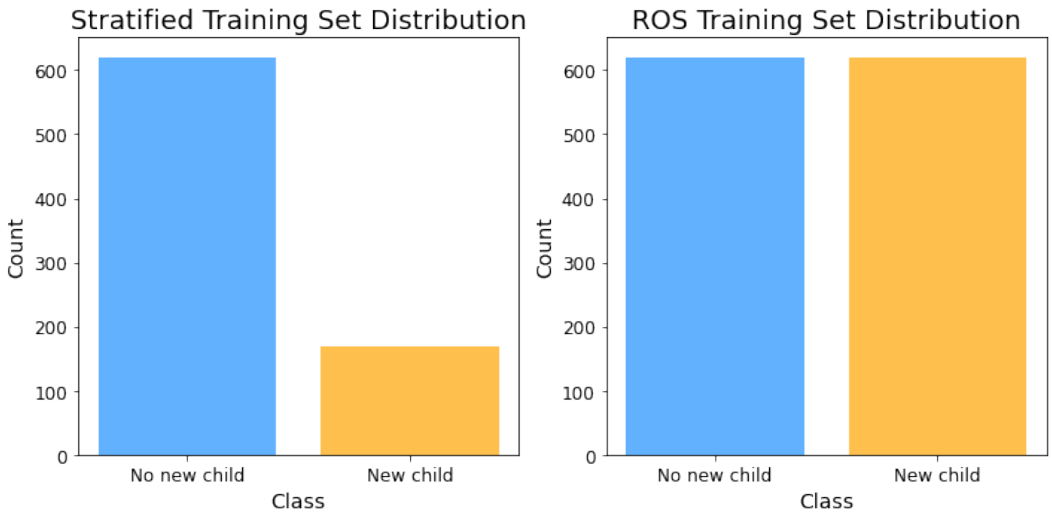

Figure 3.2: Bar plots showing the distribution of classes ("No new child" and "New child") for the 789 instances in the Stratified training set (left) and the same training set with exaggerated "New Child" due to Random Over-Sampling (ROS) (right).

During validation, models trained using these different training sets are compared to assess the impact and efficacy of the methods in enhancing the model's performance on the test split. This comparison helps determine which method better addresses class imbalance and improves overall model accuracy and reliability.

3.3 Model Selection and Training

In this section, I detail the process of selecting and training various models, followed by cross-validation and bootstrap resampling to evaluate their performance. These methods are chosen for their ability to provide robust assessments of model performance, particularly in the context of class imbalance and varying training dataset sizes. By systematically evaluating multiple models and utilizing these validation techniques, a greater understanding of the effectiveness of the models on the selected features is gained.

3.3.1 Selection of Models

In this study, a variety of common classification models from the `scikit-learn` library are employed to predict the outcome variable. The selection includes a mix of both simple and complex models to ensure a comprehensive evaluation of their performance. The chosen classification models are:

- **Logistic Regression**: A linear model that predicts the probability of a binary outcome based on input features, using a logistic function.
- **Random Forest**: An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes of the individual trees.
- **Gradient Boosting**: Another ensemble technique where models are added sequentially, each correcting errors made by the previous one, optimizing a specified loss function.
- **Support Vector Machine (SVM)**: A discriminative classifier that finds the hyperplane which best separates classes in a high-dimensional space, maximizing the margin between classes.
- **K-Nearest Neighbors (KNN)**: A non-parametric method used for classification. It assigns new data points a value based on the majority value or average of its k-nearest neighbors.
- **Gaussian Process Classifier**: A probabilistic model that defines a distribution over functions, allowing for uncertainty estimates and non-linear decision boundaries.
- **Naive Bayes**: A simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between features.
- **Neural Network**: A computational model inspired by biological neural networks. It consists of layers of interconnected neurons that process input data and learn to recognize patterns using weight and biases.

Additionally, `scikit-learn` Dummy Classifiers based on simple rules are used as a baseline to provide a point of reference. They use simple strategies such as always predicting the most frequent class (No new child) or probabilistically predicting the outcome using the distribution of the outcome variables.

3.3.2 k-Fold Cross-Validation

To assess the performance of the selected models, Stratified k-Fold Cross-Validation (SKCV) with 5 folds will be employed. This technique involves partitioning the training data into k-stratified subsets, or folds, and iteratively training the model on k-1 folds while using the remaining fold for validation. This process is repeated k times, with each fold used exactly once as the validation data[20].

The evaluation metric used for optimization will be the F1 score[1], with a focus on maximizing it, as this is the main metric used in the PreFer data challenge[13].

Furthermore, different sets of hyperparameters will be explored for each model during the SKCV process using `gridsearchCV`[21]. This approach acknowledges that different models may benefit from distinct configurations of hyperparameters to achieve optimal performance. For instance, while one model might require a larger regularization parameter to prevent overfitting, another might perform better with a different learning rate or a specific kernel type. By conducting model-specific hyperparameter tuning via grid search within SKCV, it is ensured that each model's performance is maximized under its optimal configuration, thereby improving the overall robustness and reliability of the results.

The detailed hyperparameter search spaces for each model can be found in Appendix B.

3.3.3 Bootstrap Resampling

Following each SKCV optimized model will undergo further evaluation using bootstrap resampling. Bootstrap resampling is a robust technique involving the repeated sampling of observations with replacement from the dataset to create multiple bootstrap samples [20]. These samples are utilized to estimate the variability of model performance metrics, specifically focusing on the F1 score in this study.

Bootstrap resampling will be performed across a range of dataset sizes, varying

[1]The F1 score is a metric that combines precision (the accuracy of positive predictions) and recall (the ability to correctly identify positive instances) into a single value. It provides a balanced measure of the accuracy of predicting whether someone will have a child, considering both the completeness and correctness of the predictions.

from 0.5 to 1.2 times the size of the entire dataset (987 instances), in increments of 0.02. Each size will undergo 1000 iterations, ensuring robust statistical analysis. This methodology allows for an assessment of how changes in training data size and different splits influence model performance and examines the sensitivity of models to variations in the training dataset.

As a result of bootstrap resampling, 95% confidence intervals (CI) will be computed for the F1 scores based on the bootstrap samples. These intervals provide a range of plausible values for the true model performance metrics, accounting for the variability introduced by sampling from the dataset.

4. Results

In this chapter, I present the results of the study aimed at predicting fertility trends in the Netherlands using various machine learning models. The results are structured around the key steps outlined in the methodology (Chapter 3), including feature selection outcomes, performance metrics of different models, and an analysis of class imbalance handling strategies.

4.1 Feature Selection Outcomes

Using the `RandomForestClassifier` with 1000 estimators and a Gini index threshold of 0.0025 for feature selection, as described in Section 3.1.3, resulted in the identification of 44 key attributes or a total of 48 when including different answers to the same categorical question. These key attributes have significant predictive power for determining the likelihood of having a new child. The feature importance scores from the classifier helped isolate variables that substantially contributed to the prediction model, providing insights into the factors influencing fertility trends. Before feature selection, categorical attributes were one-hot encoded to consider the impact of individual values. All selected attributes are from studies conducted in 2019-2020. Most likely due to having both high availability (sample criteria of participating in a 2019-2020 core study) and relevance due to being recent.

I manually categorized the 48 attributes into meaningful groups to facilitate a clearer understanding of their distribution. The distribution can be seen in Figure 4.1. The categories generally include demographic information, household characteristics, income and employment information, health metrics and media usage. In the following sections, I will go through each category to quickly review the selected attributes and discuss their relevance in predicting the likelihood of having a new child. The full distribution of categories is visible in Figure 4.1 and a full list explaining and giving the score for each attribute is available in Appendix A.

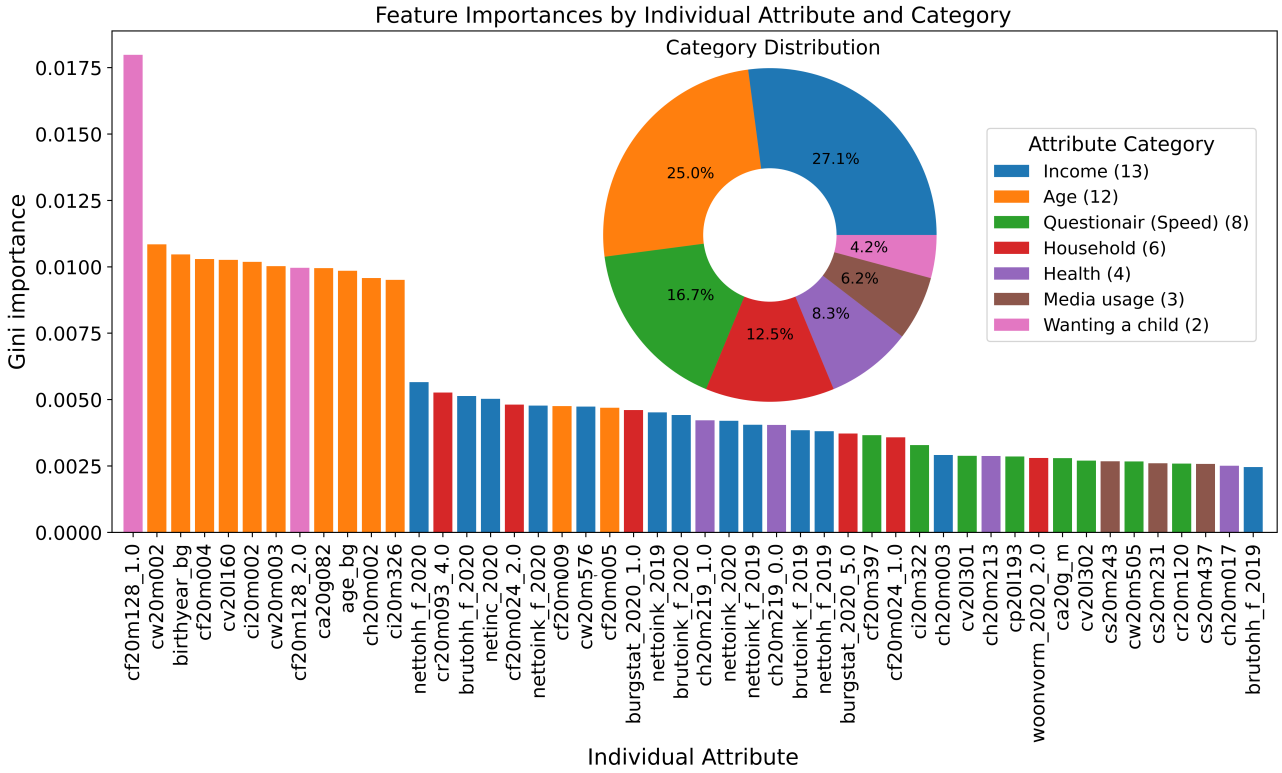

Figure 4.1: Distribution of attribute categories, for attributes selected during feature selection. The bar chart shows the gini importance for the selected attributes (codebook in Appendix A) and their respective category. The pie chart shows the relative proportions of different attribute categories, with percentages and counts displayed for each category

4.1.1 (Not) Wanting a child

The feature selection process identified several key attributes as the most influential predictors of having a new child. That said, the single most significant feature, with an importance score of ±0.018, was the response "Yes" to the question "Do you think you will have [more] children in the future?". This was by far the strongest predictor, indicating a high degree of self-awareness among respondents regarding their fertility intentions. This variable alone had a notably higher importance score compared to others, with the remaining top features (Including the answer "No") all leaning around ±0.010 scores, highlighting its critical role in forecasting fertility trends. The score difference can be seen in Figure 3.1 and Appendix A.

4.1.2 Age

Among the selected features, age-related attributes emerged as highly relevant. Multiple (sometimes duplicate) variables related to the respondent's age and birth

year were selected from different datasets with scores around 0.010. These age-related variables included not only the respondent's current age but also the individual ages of both parents, with the scores for mother and father being nearly identical at ±0.0047.

4.1.3 Income and job status

Income-related attributes also featured prominently among the selected predictors, ranking after wanting a child and Age in a range of ±0.0045. Variables related to both personal and household income from 2019 and 2020, such as net and gross monthly income, were found to be important, suggesting that economic stability and financial considerations play a crucial role in the decision to have (more) children. The preloaded variable: "paid job or not" was also selected, indicating that whether an individual has a paid job might be a significant factor in understanding family planning decisions.

4.1.4 Health

Another relatively strong predictor is the use of a gynaecologist with both answers "Yes" and "No" having score of ±0.004. In addition to the use of a gynaecologist, the importance of health-related variables is further underscored by other selected attributes. The involvement with an acupuncturist (score ±0.0029) and the respondent's weight (score ±0.0025) also emerged as notable predictors. These attributes suggest that specific health practices and general health status are influential in family planning decisions.

4.1.5 Household

Household-related variables also play a crucial role in predicting the likelihood of having a child. The most significant attribute in this category is whether the respondent speaks Dutch with their partner (score ±0.0053), specifically the answer "Not applicable". This answer might indirectly indicate the absence of a partner. Other influential variables include the respondent's current partnership status, with "No" (score ±0.0048) and "Yes" (score ±0.0036) being key indicators. Civil status variables, such as being married (score ±0.0046) and never having been married (score ±0.0037), also rank highly. Furthermore, the domestic situation, specifically (un)married co-habitation without children (score ±0.0028), is a relevant predictor. These findings suggest that marital status and cohabitation arrangements are con-

nected to family planning.

4.1.6 Media usage

Variables related to media usage, though not as highly ranked as health and household attributes, still show some relevance. The average number of hours per week spent on computer or laptop use at work (score ±0.0027), listening to music (score ±0.0026), and watching online films or TV programs (score ±0.0026) are included among the selected features. This indicates that digital engagement and media consumption patterns may have a minor influence on decisions regarding having children.

4.1.7 Questionnaire (Speed)

The duration it took respondents to complete the questionnaire is another category that emerged in the feature selection process. Multiple variables measuring duration in seconds were identified, with scores ranging from ±0.0026 to ±0.0037. The inclusion of these variables might reflect the thoroughness or decisiveness of the respondents' answers, potentially correlating with their clarity or conviction about family planning. One questionable selected attribute is the year and month of fieldwork period of an economic survey. From feature selection alone it is unclear why this might be related to the outcome variable. This variable could potentially reflect broader economic or societal conditions during the time of the survey, influencing respondents' outlook on starting or expanding a family.

4.1.8 Overview Feature Selection

The feature selection process reveals that the decision to have more children is influenced by a combination of health practices, household dynamics, media consumption, and the nature of the survey response itself. The strongest predictors are the direct inquiries about future childbearing intentions and age-related factors. Economic stability, as indicated by income and job status, also plays a significant role. This multifaceted approach underscores the complexity of fertility decisions, encompassing a range of personal, social, and economic factors.

4.2 Stratified Cross Validation Results

In this study, I utilized two different training sets: a normal stratified set and one with additional Random Over Sampling (ROS). For both training sets, I applied Stratified K-Fold Cross Validation (SKCV) with 5 folds to determine the optimal hyperparameters for each model. Additionally, I used the models to score a stratified holdout set (198 entries) and included a dummy classifier with SKCV as a baseline for comparison. The full results of the models, including accuracy, balanced accuracy, precision, recall and best hyperparameters for each model can be found in Appendix C.

4.2.1 Stratified Training Set Results

The performance of the models trained on the normal stratified set showed varying degrees of effectiveness. This is visible in Figure 4.2. The best dummy classifier, serving as the baseline, had an average SKCV F1 score of 0.2752 and a slightly higher F1 score on the test data of 0.3077, indicating its poor predictive power.

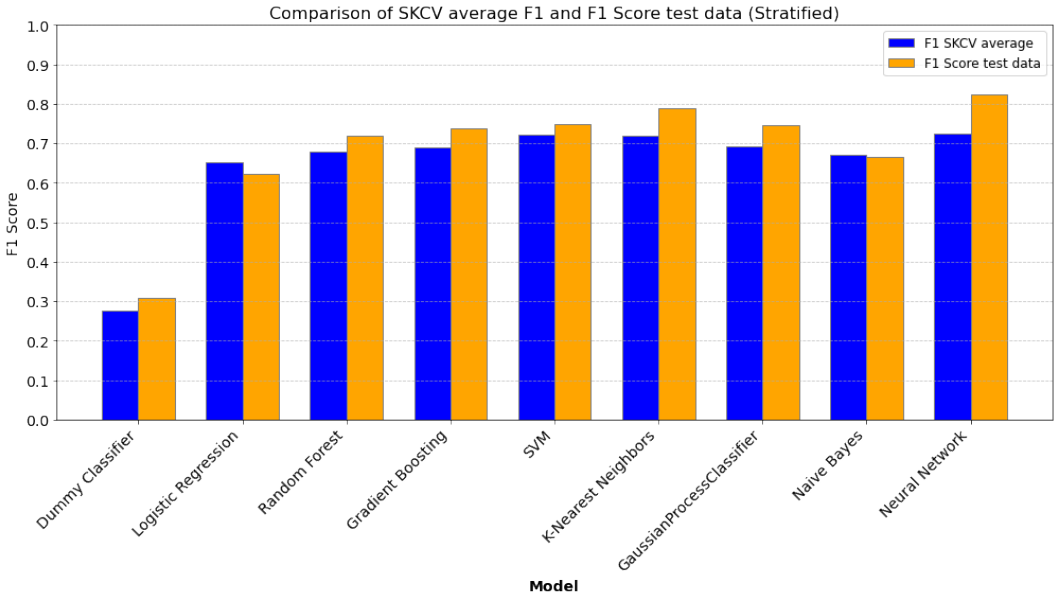

Figure 4.2: Barchart comparing average F1 score during SKCV and f1 score on test set data for each model with hyperparameters optimized on the stratified training set

Among the more sophisticated models, the Neural Network achieved the highest test F1 score of 0.8235, indicating its robust ability to capture the underlying patterns in the data. The proximity-based K-Nearest Neighbors (KNN) followed closely with a test F1 score of 0.7895. The Support Vector Machine (SVM) also per-

formed relatively well, achieving a test F1 score of 0.75, reflecting its strength in finding optimal decision boundaries.

Other models, such as GaussianProcessClassifier (test F1 score 0.7467), Gradient Boosting (test F1 score 0.7368), and Random Forest (test F1 score 0.72), also showed strong performance, confirming their capabilities in handling complex datasets. Logistic Regression (test F1 score 0.6234) and Naive Bayes (test F1 score 0.6667) had relatively lower scores but still outperformed the dummy classifier significantly, suggesting they were able to capture some meaningful relationships in the data.

4.2.2 ROS Training Set Results

Using ROS noticeably influenced the models' performance, generally resulting in slightly higher F1 scores. The dummy classifier with ROS had a test F1 score of 0.3172, remaining the least effective model.

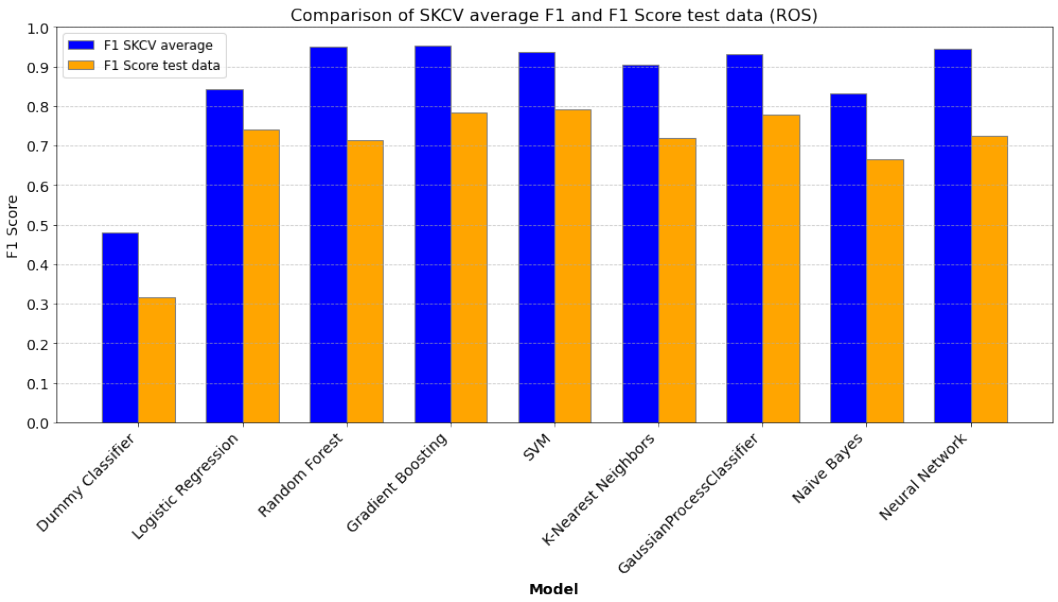

Figure 4.3: Barchart comparing average F1 score during SKCV and f1 score on test set data for each model with hyperparameters optimized on the ROS training set

As visible in Figure 4.3 the SVM model achieved the highest test F1 score of 0.7912, followed closely by Gradient Boosting at 0.7848 and the GaussianProcessClassifier at 0.7778. The Logistic Regression and Neural Network also showed strong performance, with F1 scores of 0.74 and 0.7234, respectively.

Interestingly, while the Random Forest model had one of the highest SKCV F1 score of 0.9514, its test F1 score was 0.7143, suggesting potential overfitting or sensitivity to the specific test split. Similarly, the KNN model, despite having a high SKCV F1 score of 0.9055, achieved a lower test F1 score of 0.7184 compared to its stratified version.

4.2.3 Interpretation and Comparisons

The scores on the test data are notably high, especially for the stratified test dataset when compared to SKCV, which might indicate an overestimation due to a favorable test split. This is a common issue for train/test splits, where the test set may not always be representative of the overall data distribution, by being slightly easier to predict due to less variance. This potential bias underscores the importance of using bootstrapping to validate the robustness of the model performance, as SKCV and the stratified test split provide only an initial indication of the best or near-best hyperparameters but may not fully account for variability in the data.

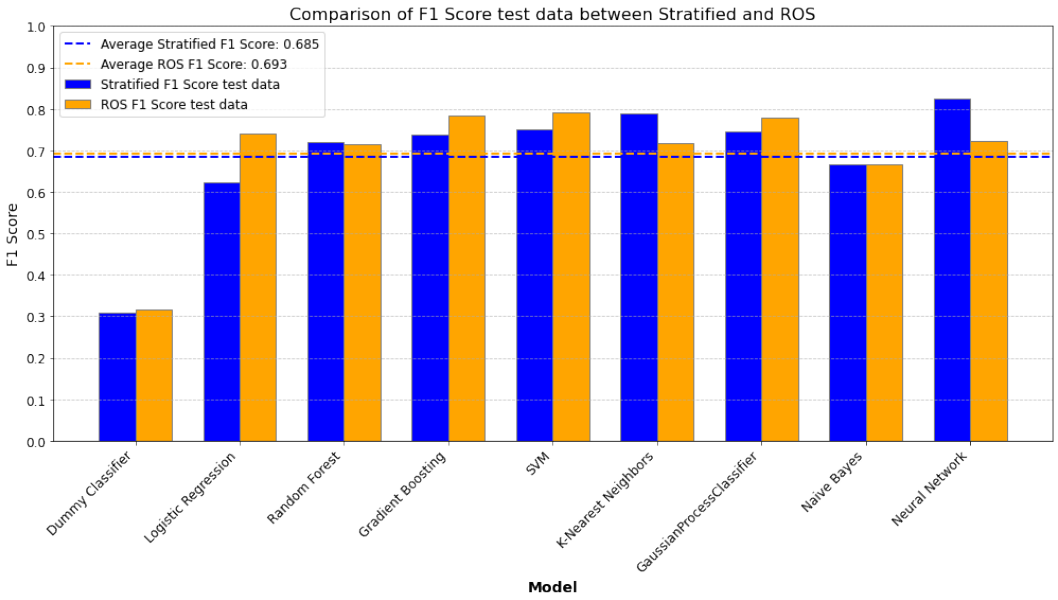

Figure 4.4: Barchart comparing performance on test set data for models optimised on either the Stratified or ROS training set. Also showing a dotted average line comparing the two methods

Comparing the results between the normal stratified and ROS techniques in Figure 4.4 reveals that ROS generally enhances model performance on the test data. The average test F1 score for all models (excluding the dummy classifier) was 0.685 for the stratified set and 0.693 for the ROS set. This suggests that addressing class im-

balance through oversampling can lead to better predictive accuracy and reliability. Most models showed an improvement or similar performance with ROS, except for the Neural Network and KNN, whose performance decreased. This highlights that while ROS can be beneficial, its impact can vary depending on the model architecture and the nature of the data.

In conclusion, SKCV is a valuable tool for hyperparameter tuning and initial model evaluation. The results indicate that ROS can improve model performance by addressing class imbalance. However, the subsequent use of bootstrapping will be essential to confirm these findings and ensure that the models' high performance is not merely due to favorable test splits or the limitations of cross-validation. This comprehensive approach will help identify the most robust and reliable models for predicting future outcomes.

4.3 Bootstrap Resampling Analysis

This section analyses the results of model performance for bootstrap resampling performed on the dataset. Bootstrap resampling involves creating a training set by randomly sampling with replacement from a training split. This was conducted 1000 times for each dataset size X, ranging from 0.5 (±500) to 1.2 (±1200) times the size of the entire dataset (987 instances). Each iteration trained the model on a new bootstrap resample of a random training split, using the best parameters identified from Section 4.2 obtained through SKCV.

Every iteration the dataset was initially randomly (not stratified) split into 80% (789 instances) training data and 20% (198 instances) test data. For each bootstrap resample, the model was trained on a sample of size X drawn with replacement from the training data and always tested on a random test set of size 198. Never seeing any test data during training.

To address class imbalance, alternatively, ROS was applied on random training splits before bootstrap resampling and training the model. This technique equalizes the distribution between the outcome variables 'No new child' and 'New child', potentially enhancing model performance as introduced in Section 3.2.2.

4.3.1 Results and model comparison

Figure 4.5 summarizes the comparative results of all models used. Overall, the models showed high F1 scores, ranging around 0.68 to 0.72. The 95% CI for the F1 scores across the 1000 runs for each step is relatively narrow, indicating reliable and consistent performance of the models. All models consistently outperformed a dummy classifier (F1 ±0.3) across the 1000 random test splits, demonstrating their predictive efficacy. A significant observation is the marked improvement in model performance when ROS was applied, evident across all dataset sizes and nearly all models. Particularly noteworthy were the Gradient Boosting, GaussianProcess-Classifier and Neural Network models, generally achieving F1 scores above 0.70 on all data sizes, indicating their overall high predictive capabilities.

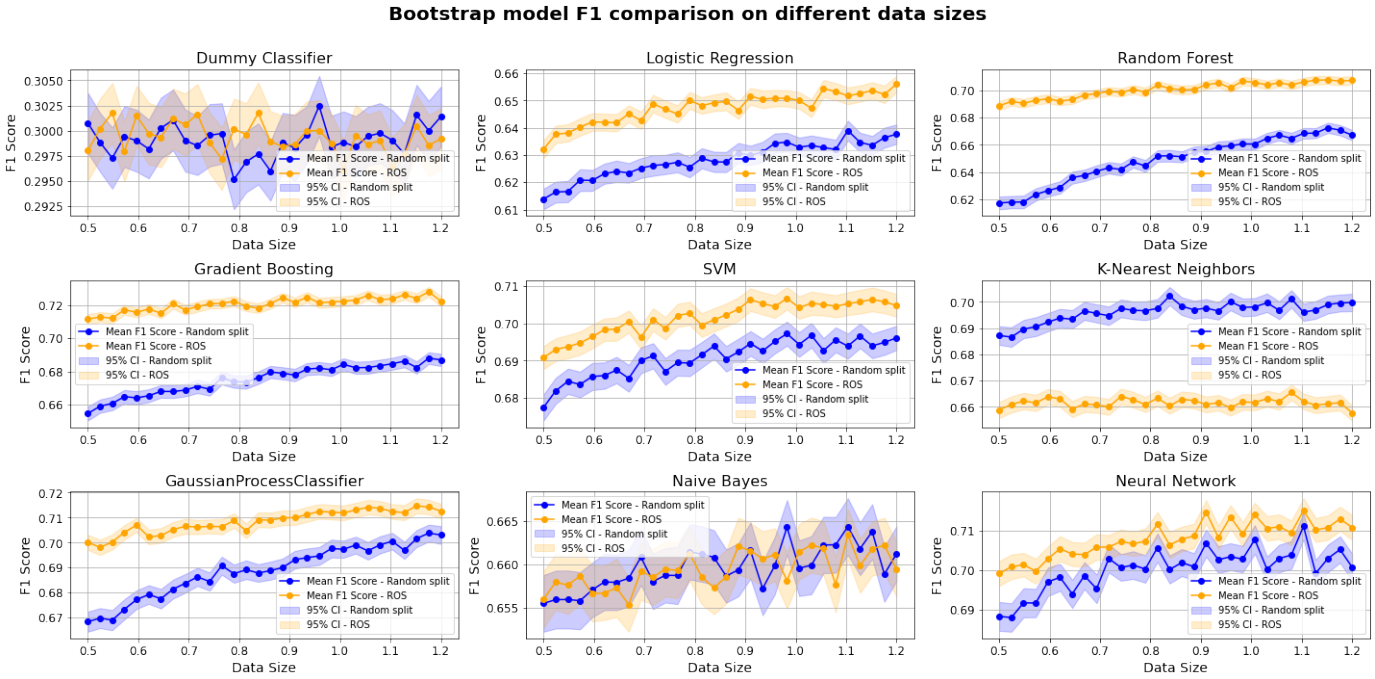

Figure 4.5: Chart comparing the mean F1 score and 95% Confidence interval (CI) for each model on different random split bootstrap data sizes while using ROS or not. Each data point represents the average result over 1000 runs each with a random train-test split of 80%-20%.

Random Forest, KNN, Gradient Boosting and GuassianProcessClassifier exhibited comparable performance across different sample sizes, suggesting robustness to variations in dataset size. Even performing relatively well on smaller training sets.

For KNN and Naive Bayes using ROS does not appear to increase performance.

For the distance-based KNN, this might be due to there being less meaningful separation between classes after oversampling, which can lead to noise and reduced effectiveness in capturing the true class boundaries. In the case of Naive Bayes, the assumption of feature independence may not hold well with the synthetic data generated by ROS, leading to suboptimal performance. These observations suggest that while ROS can be beneficial for many models, its impact can vary depending on the specific characteristics and assumptions of each model.

In conclusion, this bootstrap resampling analysis demonstrates that all evaluated models outperform the Dummy Classifier and generally benefit from ROS. The Logistic Regression and Neural Network models are particularly notable for their high performance. The stability of models like Gradient Boosting and Gaussian Process Classifier across different data sizes underscores their robustness in handling variations in training data size. However, the KNN and Naive Bayes models highlight that the benefits of ROS are not universal and depend on the nature of the model and the data. Importantly, the relatively narrow 95% confidence intervals observed across all models indicate that the models' performance is highly robust and reliable on the PreFer data.

5. Conclusion

This study aimed to predict household fertility in the Netherlands using a fully data-driven approach leveraging the large-scale longitudinal LISS dataset. By employing many processing techniques and various machine learning models such as Neural network, random forest, and linear regression classifiers, the research explored the predictive potential of the dataset and identified the most relevant socio-economic and demographic factors influencing fertility.

The feature selection process highlighted key predictors, including respondents' future childbearing intentions, age, income, job status, health metrics, household dynamics, and media usage patterns. Notably, the intention to have more children emerged as the most significant predictor, underscoring the self-awareness of individuals regarding their fertility decisions. This was followed by age and income as important predictors.

The hyperparameter evaluation of different models through Stratified K-Fold Cross Validation and the application of Random Over Sampling (ROS) to address class imbalance revealed insightful results. Models like Neural Networks, Support Vector Machines, and Gradient Boosting demonstrated strong performance with high F1 scores (0.7+), indicating their robust ability to capture underlying patterns in the data. The results also underscored the benefits of ROS in improving model performance, although its impact varied across different models.

Bootstrap resampling further validated the consistency and reliability of the models. The analysis confirmed that all evaluated models well outperformed the dummy classifier, with particularly high performance observed in Gradient Boosting and Gaussian Process Classifiers. The stability of these across different data sizes highlighted their robustness in handling variations in training data.

While the current results are promising, there remains significant potential for further refinement and expansion. Future research could explore additional variables

and the aggregation of surveys, more advanced machine learning techniques, or integration with other data sources to improve predictive accuracy and broaden the scope of insights generated. This ongoing development underscores the dynamic nature of AI-driven fertility prediction and highlights the need for continued innovation.

In summary, this research successfully demonstrated the potential of AI-driven methods to predict fertility trends using the LISS dataset. The findings contribute to the PreFer data challenge by enhancing the understanding of predictive modeling of fertility. By showcasing the effectiveness of data-centric methodologies, this study offers a framework that can be adapted and applied to other countries and datasets or change the focus questions during data collection, potentially leading to global improvements in fertility prediction and policy planning.

6. Discussion

This research presents a data-driven approach to predict fertility trends in the Netherlands using machine learning models and the extensive longitudinal LISS dataset. The study identified and processed relevant attributes, then evaluated various prediction models. The results demonstrate the potential of AI-driven methods in fertility prediction, but several methodological choices and limitations warrant further discussion.

Dataset size

A critical discussion point is the limited number of outcome labels available for analysis, totaling 987 in this study. Given the strong predictive influence observed for variables such as desire for a child, age, and income, there remains a need for a larger dataset to accurately assess the predictive power of the other selected attributes. The smaller sample size of outcome labels may have influenced the ability to fully capture the potential predictive value of lesser-known factors or less frequently occurring variables. Increasing the dataset size could provide more robust insights into the relative importance and contribution of these additional attributes towards predicting fertility outcomes.

Feature selection on dataset

One key methodological choice was the selection of attributes with a high availability (75%+). This approach was deemed appropriate based on the analysis in section 2.4. However, this decision might have overlooked valuable predictive information due to not directly taking into account surveys that have been performed over multiple waves, thus having the same question count as a different attribute reducing the availability of the question on an attribute level. This means some survey questions might not have been selected, simply due to the survey being split into multiple smaller waves. Future research could benefit from first aggregating these survey questions spanning multiple waves to one question variable to include or reveal new questions and interest areas with strong predictive power. This would ensure a more comprehensive analysis.

The selected attributes in this study represent only the tip of the iceberg, by only including the ones that were easily available within the dataset, serving as a proof of concept for the potential of data-driven fertility prediction. This is also the case for filtering or aggregating identical attributes that are available via multiple surveys. For instance, during feature selection, the model uses multiple similar age and income attributes which could be simplified to make more way for other non-duplicate attributes.

Preprocessing steps

As an alternative method lagged time attributes were only experimented with but not included in the main study. These attributes, which consider past values and changes within them to predict future outcomes, could significantly enhance the model's predictive power. Future research should explore the integration of lagged variables to capture temporal dynamics more effectively.

Another important consideration is data imputation. In this study, all data was imputed together, which, while avoiding data leakage, could impact accuracy. Ideally, the entire data processing pipeline should be tested within the bootstrap resampling framework to ensure robustness, by imputing test data on an imputer fitted on training data, rather than all data.

Scores and bias with small sample size

While bootstrap resampling was used to mitigate potential biases and check for robustness, the model scores might still be exaggerated due to the small sample size or underlying bias in the data. These scores should be seen as an indication of what is possible within the current dataset, rather than definitive performance metrics. Further testing with independent datasets is necessary to validate these findings. SKCV provided an initial indication of the best or near-best hyperparameters, but bootstrap resampling was the real test of model performance.

These methodological considerations and limitations highlight the complexity of data-driven fertility prediction and the used dataset, and underscore areas for future research. While this study demonstrates the potential of AI-driven approaches, it also reveals the need for ongoing refinement and validation of these methods.

7. Acknowledgement

In this paper, I make use of data from the LISS panel (Longitudinal Internet studies for the Social Sciences) managed by Centerdata (Tilburg University, The Netherlands) as provided in the context of the PreFer Data Challenge.

A. Feature selection

Below are the detailed tables of the selected features, categorized by their importance scores and grouped by relevant attributes such as health, household, income, job status, media usage, and questionnaire speed. First Figure A.1 showing the difference in gini importance for the selected attributes.

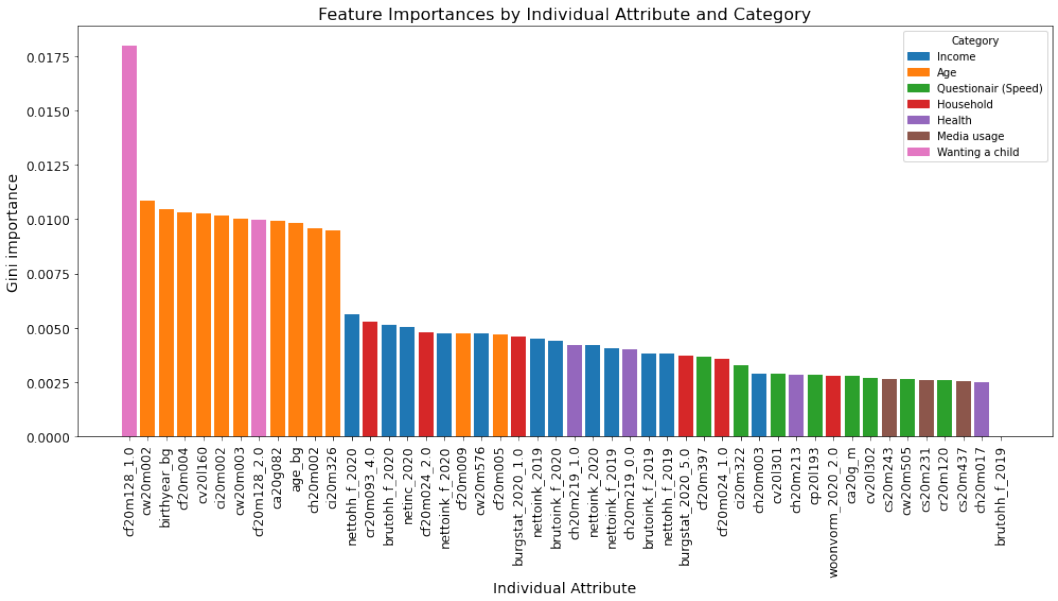

Figure A.1: Barchart of gini importance for selected attributes

Table A.1: Feature Selection for Predicting Childbirth

individual_attribute	score	var_label	categorical_-value	Category
cf20m128_1.0	0.017982	Do you think you will have [more] children in the future?	Yes	Wanting a child
cf20m128_2.0	0.009958	Do you think you will have [more] children in the future?	No	Wanting a child

Continued on next page

Table A.1 continued from previous page

individual_attribute	score	var_label	categorical_-value	Category
cw20m002	0.010841	Respondent's year of birth	-	Age
birthyear_bg	0.010467	Year of birth [imputed by PreFer organisers]	-	Age
cf20m004	0.010291	Preload variable: Age respondent	-	Age
cv20l160	0.01026	Preloaded variable: age - part 2	-	Age
ci20m002	0.010187	Age respondent	-	Age
cw20m003	0.010025	Respondent's age	-	Age
ca20g082	0.009948	Age respondent	-	Age
age_bg	0.00985	Age of the household member on December 2020 [imputed by PreFer organisers]	-	Age
ch20m002	0.009577	preloaded variable: age	-	Age
ci20m326	0.009505	Year of birth respondent	-	Age
cf20m009	0.004753	What is the year of birth of your mother?	-	Age
cf20m005	0.004688	What is the year of birth of your father?	-	Age
ch20m219_1.0	0.004216	gynaecologist	Yes	Health
ch20m219_0.0	0.004043	gynaecologist	No	Health
ch20m213	0.002873	acupuncturist	-	Health
ch20m017	0.002506	How much do you weigh, without clothes and shoes?	-	Health
cr20m093_4.0	0.005266	Do you speak Dutch with…your partner?	"not applicable"	Household
cf20m024_2.0	0.004812	Do you currently have a partner?	No	Household

Continued on next page

Table A.1 continued from previous page

individual_attribute	score	var_label	categorical_-value	Category
burgstat_2020_1.0	0.004606	Civil status	Married	Household
burgstat_2020_5.0	0.00372	Civil status	Never been married	Household
cf20m024_1.0	0.003576	Do you currently have a partner?	Yes	Household
woonvorm_2020_2.0	0.002802	Domestic situation	(Un)married co-habitation, without child(ren)	Household
nettohh_f_2020	0.005655	Net household income in Euros	-	Income
brutohh_f_2020	0.00513	Gross household income in Euros	-	Income
netinc_2020	0.00503	Personal net monthly income in Euros	-	Income
nettoink_f_2020	0.004771	Personal net monthly income in Euros, imputed	-	Income
cw20m576	0.004738	Current income per month, based on values from the Core Questionnaire Income	-	Income
nettoink_2019	0.004515	Personal net monthly income in Euros (incl. nettocat)	-	Income
brutoink_f_2020	0.004419	Personal gross monthly income in Euros, imputed	-	Income
nettoink_2020	0.004197	Personal net monthly income in Euros (incl. nettocat)	-	Income

Continued on next page

Table A.1 continued from previous page

individual_attribute	score	var_label	categorical_-value	Category
nettoink_f_2019	0.004048	Personal net monthly income in Euros, imputed	-	Income
brutoink_f_2019	0.003845	Personal gross monthly income in Euros, imputed	-	Income
nettohh_f_2019	0.003805	Net household income in Euros	-	Income
brutohh_f_2019	0.003800	Gross household income in Euros	-	Income
ch20m003	0.002911	preloaded variable: paid job or not	-	Income
cs20m243	0.002678	computer or laptop use, average number of hours per week: at work	-	Media usage
cs20m231	0.002602	listening to music, average time expenditure on days that apply, hours	-	Media usage
cs20m437	0.002577	average number of hours per week spent on: watching online films or TV programs	-	Media usage
cf20m397	0.003657	Duration in seconds	-	Questionnaire (Speed)
ci20m322	0.003285	Duration in seconds	-	Questionnaire (Speed)
cv20l301	0.002879	Duration in seconds - part 1	-	Questionnaire (Speed)
cp20l193	0.002855	Duration in seconds	-	Questionnaire (Speed)
ca20g_m	0.002794	Year and month of field work period	-	Questionnaire (Speed)

Continued on next page

Table A.1 continued from previous page

individual_attribute	score	var_label	categorical_-value	Category
cv20l302	0.0027	Duration in seconds - part 2	-	Questionnaire (Speed)
cw20m505	0.002666	Duration in seconds	-	Questionnaire (Speed)
cr20m120	0.002586	Duration in seconds	-	Questionnaire (Speed)

B. Model hyperparameter search spaces

The hyperparameter settings used when performing gridsearchCV, which tries all combinations of given parameters for each model

- **Logistic Regression**
  - **Parameters:**

```
C: [0.001, 0.01, 0.1, 1.0, 10.0, 100.0]
penalty: ['l1', 'l2']
solver: ['liblinear', 'saga']
```

- **Random Forest**
  - **Parameters:**

```
n_estimators: [100, 300, 500, 800, 1000]
max_depth: [None, 10, 30, 50]
min_samples_split: [2, 5, 10]
min_samples_leaf: [1, 4, 8]
max_features: ['auto', 'sqrt', 'log2']
```

- **Gradient Boosting**
  - **Parameters:**

```
n_estimators: [100, 500]
learning_rate: [0.01, 0.1]
max_depth: [3, 5]
subsample: [0.5, 1.0]
min_samples_split: [2, 5]
min_samples_leaf: [1, 2]
max_features: ['auto', 'sqrt']
```

- **SVM**

- **Parameters:**

```
C: [0.1, 1.0, 10.0, 100.0]
kernel: ['linear', 'poly', 'rbf', 'sigmoid']
gamma: ['scale', 'auto']
degree: [2, 3, 4, 5]
```

- **K-Nearest Neighbors**
  - **Parameters:**

```
n_neighbors: [3, 5, 7, 9, 11, 13, 15, 17, 19]
weights: ['uniform', 'distance']
algorithm: ['auto', 'ball_tree', 'kd_tree', 'brute']
```

- **GaussianProcessClassifier**
  - **Parameters:**

```
max_iter_predict: [100, 200, 300, 400, 500, 1000]
```

- **Naive Bayes**
  - No hyperparameters to tune.
- **Neural Network**
  - **Parameters:**

```
hidden_layer_sizes: [(50,), (100,), (200,), (100, 100)]
activation: ['logistic', 'tanh', 'relu']
solver: ['lbfgs', 'adam']
alpha: [0.0001, 0.001, 0.01]
learning_rate: ['constant', 'adaptive']
batch_size: [32, 64]
beta_1: [0.9, 0.95]
beta_2: [0.999, 0.9999]
```

C. SKCV results

Below is the table of the full SKCV results and selected attributes for each model with the highest average SKCV F1 score.

Table C.1: SKCV results for each model and training set

Model	**Training set**	**F1 SKCV**	**F1 test**	**Accuracy**	**B_-Accuracy**	**Precision**	**Recall**	**Params**
Neural Network	Stratified	0.7247	0.8235	0.9242	0.8844	0.8333	0.8140	{'activation': 'tanh', 'alpha': 0.01, 'batch_size': 32, 'beta_-1': 0.9, 'beta_2': 0.999, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
SVM	Stratified	0.7219	0.7500	0.8990	0.8263	0.8108	0.6977	{'C': 100.0, 'degree': 2, 'gamma': 'auto', 'kernel': 'rbf'}
K-Nearest Neighbors	Stratified	0.7183	0.7895	0.9192	0.8392	0.9091	0.6977	{'algorithm': 'auto', 'n_neighbors': 17, 'weights': 'uniform'}
Gaussian Process Classifier	Stratified	0.6931	0.7467	0.9040	0.8127	0.8750	0.6512	{'max_iter_predict': 100}
Gradient Boosting	Stratified	0.6893	0.7368	0.8990	0.8095	0.8485	0.6512	{'learning_rate': 0.01, 'max_-depth': 3, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_-samples_split': 2, 'n_estimators': 500, 'subsample': 1.0}
Random Forest	Stratified	0.6800	0.7200	0.8939	0.7978	0.8438	0.6279	{'max_depth': 10, 'max_features': 'log2', 'min_samples_-leaf': 1, 'min_samples_split': 10, 'n_estimators': 100}

Table C.1 continued from previous page

Model	Training set	F1 SKCV	F1 test	Accuracy	B_-Accuracy	Precision	Recall	Params
Naive Bayes	Stratified	0.6699	0.6667	0.8283	0.8147	0.5763	0.7907	{}
Logistic Regression	Stratified	0.6511	0.6234	0.8535	0.7468	0.7059	0.5581	{'C': 100.0, 'penalty': 'l1', 'solver': 'liblinear'}
Dummy Classifier	Stratified	0.2752	0.3077	0.5000	0.5042	0.2200	0.5116	{'strategy': 'uniform'}
Gradient Boosting	ROS	0.9533	0.7848	0.9141	0.8443	0.8611	0.7209	{'learning_rate': 0.1, 'max_-depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_-samples_split': 2, 'n_estimators': 500, 'subsample': 1.0}
Random Forest	ROS	0.9514	0.7143	0.8788	0.8134	0.7317	0.6977	{'max_depth': None, 'max_-features': 'log2', 'min_samples_-leaf': 1, 'min_samples_split': 2, 'n_estimators': 1000}
Neural Network	ROS	0.9437	0.7234	0.8687	0.8405	0.6667	0.7907	{'activation': 'relu', 'alpha': 0.001, 'batch_size': 32, 'beta_1': 0.95, 'beta_2': 0.9999, 'hidden_-layer_sizes': (100, 100), 'learning_rate': 'constant', 'solver': 'adam'}
SVM	ROS	0.9369	0.7912	0.9040	0.8799	0.7500	0.8372	{'C': 10.0, 'degree': 2, 'gamma': 'scale', 'kernel': 'rbf'}
Gaussian Process Classifier	ROS	0.9306	0.7778	0.8990	0.8683	0.7447	0.8140	{'max_iter_predict': 100}
K-Nearest Neighbors	ROS	0.9055	0.7184	0.8535	0.8560	0.6167	0.8605	{'algorithm': 'auto', 'n_neighbors': 3, 'weights': 'distance'}
Logistic Regression	ROS	0.8428	0.7400	0.8687	0.8657	0.6491	0.8605	{'C': 100.0, 'penalty': 'l2', 'solver': 'saga'}

Table C.1 continued from previous page

Model	Training set	F1 SKCV	F1 test	Accuracy	B_-Accuracy	Precision	Recall	Params
Naive Bayes	ROS	0.8314	0.6667	0.8182	0.8251	0.5538	0.8372	{}
Dummy Classifier	ROS	0.4806	0.3172	0.5000	0.5126	0.2255	0.5349	{'strategy': 'stratified'}

Bibliography

[1] DE Bloom. "Demographics can be a potent driver of the pace and process of economic development". In: *Finance and Development Journal* (2020), p. 6.

[2] Hans-Peter Kohler, Francesco C Billari, and José Antonio Ortega. "The emergence of lowest-low fertility in Europe during the 1990s". In: *Population and development review* 28.4 (2002), pp. 641–680.

[3] Nicoletta Balbo, Francesco C Billari, and Melinda Mills. "Fertility in advanced societies: a review of research: La fécondité dans les sociétés avancées: un examen des recherches". In: *European Journal of Population/Revue europeenne de demographie* 29 (2013), pp. 1–38.

[4] John Bongaarts and Griffith Feeney. "On the quantum and tempo of fertility". In: *Population and development review* (1998), pp. 271–291.

[5] Tomáš Sobotka. "Childlessness in Europe: Reconstructing long-term trends among women born in 1900–1972". In: *Childlessness in Europe: Contexts, causes, and consequences* (2017), pp. 17–53.

[6] Joshua R Goldstein, Tomáš Sobotka, and Aiva Jasilioniene. "The end of "lowest-low" fertility?" In: *Population and development review* 35.4 (2009), pp. 663–699.

[7] Tomáš Sobotka. "Is lowest-low fertility in Europe explained by the postponement of childbearing?" In: *Population and development review* 30.2 (2004), pp. 195–220.

[8] Gary S Becker. "An economic analysis of fertility". In: *Demographic and economic change in developed countries*. Columbia University Press, 1960, pp. 209–240.

[9] John Bongaarts. "A framework for analyzing the proximate determinants of fertility". In: *Population and development review* (1978), pp. 105–132.

[10] John Bongaarts and Susan Cotts Watkins. "Social interactions and contemporary fertility transitions". In: *Population and development review* (1996), pp. 639–682.

[11] Roberta Rocca and Tal Yarkoni. "Putting psychology to the test: Rethinking model evaluation through benchmarking and prediction". In: *Advances in Methods and Practices in Psychological Science* 4.3 (2021), p. 25152459211026864.

[12] Tal Yarkoni and Jacob Westfall. "Choosing prediction over explanation in psychology: Lessons from machine learning". In: *Perspectives on Psychological Science* 12.6 (2017), pp. 1100–1122.

[13] Elizaveta Sivak et al. "Combining the strengths of Dutch survey and register data in a data challenge to predict fertility (PreFer)". In: *Journal of Computational Social Science* (2024), pp. 1–29.

[14] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3 (2011), pp. 1–67. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

[15] Lucas BV de Amorim, George DC Cavalcanti, and Rafael MO Cruz. "The choice of scaling technique matters for classification performance". In: *Applied Soft Computing* 133 (2023), p. 109924.
[16] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.
[17] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.
[18] *Forest Importances Example Sklearn*. Accessed: 2024-06-19. 2024. URL: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html.
[19] Ajinkya More. "Survey of resampling techniques for improving classification performance in unbalanced datasets". In: *arXiv preprint arXiv:1608.06048* (2016).
[20] Ron Kohavi et al. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai*. Vol. 14. 2. Montreal, Canada. 1995, pp. 1137–1145.
[21] *GridSearchCV Sklearn*. Accessed: 2024-06-19. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.