Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics and Management Science - Quantitative Finance

Evaluating Imputation Techniques and Imputation Uncertainty for Borrower Characteristics in Risk Assessment for Dutch Interest-Only Mortgages

Wilco Koomen (545486)

Supervisor:	dr. Wendun Wang
Second assessor:	dr. Nick Koning
Internship Company:	PwC
Company Supervisor:	Sharon Verschelling
Second Company Supervisor:	Peter Spoelstra
Date final version:	10th August 2024

Abstract

Interest-only mortgages (IOM) form a significant portion of the Dutch mortgage market, making up around half of all mortgages. These mortgages inherently carry risks, mainly due to the lack of ongoing borrower information post-application. This research addresses these challenges by exploring various imputation methods to handle missing borrower data, aiming to fill in missing values in current datasets. Methods like Ridge regression, Multiple Imputation (MI), K-Nearest Neighbours (KNN), and Random Forest (RF) are evaluated as well as an imputation combination method. Additionally, the research investigates the uncertainty around imputed the values using a bootstrapping technique and checks the impact of the resulting uncertainty on risk modelling.

The findings indicate that the RF method provides the most accurate point estimates for imputations. Combining it with multiple methods in a Relative Score combined imputation leads to a further increase in performance, highlighting the advantage of combination approaches in data imputation. The study finds substantial differences in imputation variance between the different imputation methods, with the RF and KNN methods yielding the narrowest confidence intervals, but these methods also show higher rates of confidence interval violation, potentially highlighting a poor estimation of the imputation variance.

The research also investigates the implications of imputation uncertainty on default probability predictions, revealing that while the impact is statistically significant, it remains relatively small in practical terms. This shows the suitability of data imputation in imputing borrower characteristics for IOM's.

Contents

1 Introduction

In the Dutch mortgage market, interest-only mortgages (IOM) make up for a significant portion of mortgages, comprising 49% of all mortgages (European Systemic Risk Board, 2022). However, these mortgages contain inherent risks, for an important part due to the lack of ongoing borrower information after the application of the mortgage. Borrowers are not required to provide updates on their income, financial position, or repayment ability. Combined with recent economic shifts, this poses considerable challenges. Rising consumer prices, soaring energy costs, and fluctuating interest rates strongly impact the affordability of repayment for the borrower, leading to uncertainty on whether they are able to meet lump-sum repayments or to refinance the current outstanding mortgage at maturity. Moreover, a substantial number of these mortgages are set to mature in the next 10-15 years (Autoriteit Financiële Markten (AFM), 2023), while simultaneously projections foresee a potential decline in housing prices (Sylvain Broyer, 2023). This subsequently decreases the value of the collateral of the loan, and with repayment by the borrower being uncertain, collateral sale might be a main mean of repayment. These factors combined make the Dutch mortgage market face risks from multiple aspects.

The combination of these factors presents a set of challenges related to each other. Firstly, the incomplete information on property valuation, together with anticipated declines in house prices, threatens instability and potential disruptions in the housing market, due to forced sale to meet repayment needs. Secondly, rising living costs and uncertainties regarding property values could lead to confusion among borrowers, making it more difficult for them to come up with options to meet repayment needs at maturity. Thirdly, the vulnerability of banks to incomplete repayments at maturity highlights the need for an increase in capital reserves, which will hurt profitability in the long term. Addressing these complexities is further complicated by data-related obstacles. Existing data sources are often incomplete or outdated, leading to problems with current probability of default predictions. Providing imputation methods to come up with a more up-to-date dataset while still complying to General Data Protection Regulation (GDPR) regulations is one of the key challenges that this research aims to overcome.

To mitigate these risks and uncertainties, an approach from multiple aspects is needed. From an econometric perspective, advanced imputation methods that incorporate the dynamic heterogeneity between borrowers and loans, potentially also including machine learning algorithms, can prove to be a useful addition. The call for these types of solutions is also present in the current banking landscape, with large Dutch banks looking for solutions to the incomplete and outdated client data, which is needed to quantify the IOM risk. This call for a solution is further fuelled by regulation changes by the European Central Bank (ECB). This regulation changes state that banks who hold outstanding mortgages which are either in full or in part IO, should gradually place these mortgages into a higher stage of provisioning if the last information of the borrower is older than three years, starting from the end of 2024. Increasing provisioning from stage 1 to stage 2 of provisioning leads to using a lifetime probability of default (PD) in loss calculations instead of a 12-month ahead one. This could lead to an increase in the expected loss with a factor 10-15, substantially increasing provisioning as is also shown in Figure 1, which shows the large increase in provisioning when large portions of borrower information are missing, stressing the need for adequate methods to address the challenge of missingness.

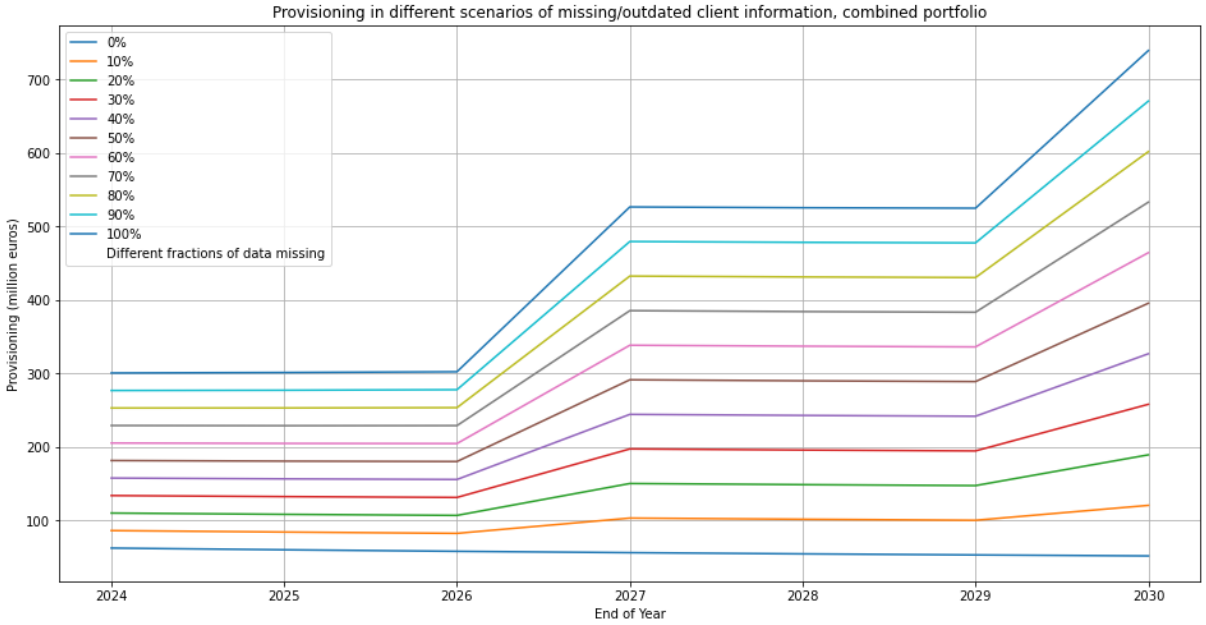

Figure 1: Example calculation of provisioning increase in case of missing information, based on a large dutch bank with a total interest only portfolio of around 60 billion, a 12-month PD of 0.659% and a 85% recovery rate

These challenges combined highlight the need for robust risk modelling methods for these interest only mortgages that can handle the missing values and uncertainty in current borrower characteristics. This uncertainty immediately poses one of the key problems with data imputation. The problem with imputed values is that they carry a certain uncertainty in the imputed value. Just taking the imputed value as a given and ignoring its uncertainty can lead to bias and invalid statistical inference. This research aims to overcome that problem by looking at ways to come up with a confidence interval around this (mean) imputed value.

By comprehensively addressing these challenges, it will allow banks with outstanding interest only mortgages to quantify risks, ensuring the stability of the housing sector and profitability of financial institutions in the long term. Therefore, this research aims to come up with the most suitable method for imputing missing loan and borrower characteristics, among multiple individual methods and a combination of these individuals, while also addressing the uncertainty that follows from the imputation. From this, the following research question is deducted:

What is the impact of various imputation methods on assessing Interest Only Mortgage risk, and how can imputation uncertainty be quantified and integrated into risk modelling frameworks?

The research finds that the Random Forest model yields the best performance in terms of point estimates, followed by the K-Nearest Neighbours and Ridge methods and finally the Multiple Imputation method. The combination of multiple imputation methods does lead to more accurate imputations than each of the individuals. The study also finds that the estimated uncertainty of the imputations for the different methods varies substantially, with the RF and KNN methods having the lowest amount of uncertainty. Although, these methods also show more violations of the constructed confidence intervals. The final finding is that the implication of the calculated uncertainty on risk modelling is significant, while still being small in practical application.

The proposed research question will be answered following the following structure in this research. Firstly, the existing research on the topic is explored, to find a gap in this literature, in Section 2. Nextly, the data that will be used for this research is presented in Section 3, this is followed by the proposed methods in Section 4. Lastly the findings of the research and concluding remarks are presented in Sections 5 and 6 respectively.

2 Literature

Even though the field of data imputation for risk factors of interest only mortgages specifically is not researched extensively, there are closely related fields that are researched in detail. These fields include but are not limited to: risk analysis in other types of mortgages, data imputation in financial research in general and data imputation in other fields like medical data. The findings in these fields can prove to be useful in the context of this research.

2.1 Background literature

The first field of research, that is relevant for this study, is existing research on mortgage risk in general, which are not necessarily interest only mortgages. For this widened scope, there are several earlier studies to be found, some dating back multiple decades, such as the research by Jackson and Kaserman (1980). This research studies driving factors of mortgage default decisions in the United States. The study finds that the term to maturity, interest rate and Loan-to-Value ratio (LTV) at origination all have a positive impact on the probability of a default decision by the borrower in their simple regression framework. On one hand this is very valuable to this research as well, as all these factors are known to the lender and need not to be imputed. However, it is worth noting that the paper by Jackson and Kaserman (1980) focuses on default decision, and voluntary default decisions are not possible in the Netherlands (Wetten.nl-Regeling-Faillisementswet, 2024). Despite the fact that the research is not directly applicable in the context of Dutch IOM's, it still provides valuable insights in driving risk factors of mortgage default risk. More recent research has provided more extensive and specific insights in this field. Research by Campbell and Cocco (2015), adds expected risks to labour income as well as macro-economic variables like house prices and inflation to a rational life cycle model. This proposed method therefore also incorporates the dynamic nature of a mortgage contract, where the characteristics of both the borrower (income growth rate) and the economy (house prices, interest rates, etc.) change over time. This research finds that borrowers with a higher expected income growth have a higher risk of default. Even though this paper touches on the dynamic nature of mortgage risk, it still assumes that borrowers can voluntarily choose to default, which is not the case for Dutch IOM's.

Contrary to the US, the United Kingdom (UK) does not allow for a so called "walk-away" option for mortgage default (Aron & Muellbauer, 2016), making it similar to the Dutch case. A study that looks at mortgage default risk in the UK is the paper by Lambrecht, Perraudin and Satchell (1997). Opposed to earlier discussed research, the research of Lambrecht et al. (1997) uses a hazard model in which borrower and mortgage characteristics are used to explain the time to default of a specific mortgage. This is done by only including defaulting mortgages into

the analysis. Included variables in the used Weibull duration models are: Loan-to-Value ratio, salary, marital status and interest rate, all at time of origination of the mortgage. The research finds that generally speaking, "ability to pay" variables have a more pronounced effect on the duration to default, with a higher salary and marriage both increasing the duration to default. The study finds that the effect of the original Loan-to-Value (LTV) ratio has an insignificant but positive effect on the duration till default, translating to a decrease in default risk. This contradicts earlier literature, such as Jackson and Kaserman (1980), who find that the LTV ratio has a positive effect on default risk. Part of this can potentially be explained through a common practice since the 1980s of taking out second mortgages, to finance other purchases than a house. These loans typically have low ratios of the loan value to the property value, but the risks are generally high (Lambrecht et al., 1997). Research by Burrows (1998) further explores this field. This research explains the odds of being in mortgage arrears through a logistic regression on a variety of borrower and loan characteristics. The research is performed on UK mortgages on a micro level. Some drivers that have been found to affect mortgage risk are: age (young borrowers pose a larger risk), divorces, current employment status, social class, self-employment, household structure, Loan-to-Value and, especially interesting for this research, type of mortgage. In this case, Interest Only mortgages are found to be less risky than Capital and Interest mortgages. Although it is worth noting here that this research focuses on three month mortgage arrears during the term of contract. The risks associated with IOM's typically entail risks at maturity and not during the contract. During the contract, monthly payments for IOM's are usually lower, because no repayment is done during the term, potentially decreasing the risk of mortgage arrears. Research by de Haan and Mastrogiacomo (2020) does find that IOM's carry more risk when looking at non-performance in general. This is found in a similar framework as Burrows (1998), in a logistic regression with borrower and loan characteristics as explanatory variables, this is performed on a Dutch loan level dataset, making this research particularly interesting for this research. Again, the age of the applicant is deemed to have a negative effect on mortgage risk, together with the presence of a national mortgage insurance contract (Nationale Hypotheek Garantie). Other factors like LTV, self-employment and Debt service-to-Income have been found to have a positive effect on mortgage non-performance. Especially the Debt service to Income (DSTI) variable is interesting since the effect is large and highly significant. However, the research by de Haan and Mastrogiacomo (2020) treats the included variables like income as given, which is not necessarily the case in the bank database that will be used in this research. This leads to the need of imputation of these variables if they are required to be included in the risk analysis.

2.2 Data imputation

In the context of Dutch IOM's, a large proportion of these risk driving factors are either outdated or missing entirely. Therefore, this research aims to propose a sensible way of imputing the missing values that would be needed for the mortgage risk analyses as described in earlier literature. This research field of data imputation is one that has been researched extensively, stretching to, but not being limited to imputing variables that are relevant for mortgage default predictions and are missing at maturity of these mortgages.

One paper that investigates solving this issue through linear regression is Dombrowski, Pace and Wang (2022). This research imputes the heterogenous dynamics in borrower characteristics of mortgages. Due to the large number of possible predictors, a Ridge regression method is used to avoid multicollinearity. The paper finds the proposed Ridge framework to work well in imputing borrower specific effects out-of-sample.

When considering a wider scope, imputation methods are used in all sorts of fields, a potentially surprising one being the National Health Interview Survey (NHIS) (Schenker et al., 2006). Even though a survey on medical information seems very far off from borrower characteristics in IOM's, the methods that are used in one application could pose useful for the other as well. For example, in both applications, the income of a household is one of the key variables of interest, while the missingness is large (low response rate in surveys for the NHIS and limited info for the bank in case of IOM's). The paper by Schenker et al. (2006) employs a Multiple Imputation (MI) method to impute these missing variables like household income. This methodology is found to work well in this case, even correcting for biases that occur in estimates based on the data without imputation. Another important finding is that Multiple Imputation results in gains in efficiency as well, which makes it a suitable method when looking at the uncertainty of imputations.

Missing values are a more common problem in the medical world, this becomes apparent from the large number of studies that focus on missing value imputation in medical data. Another of which is the research by Wang, Tang, Wu, Wang and Zhang (2022) who look at the most suitable way of imputing missing values in the context of clinical decision making and find that especially an Ensemble learning (EL) but also a Random Forest (RF) imputation method work well in this context. Even though these imputed variables do not necessarily resemble those of risk in IOM's, it could prove to be a useful addition to also include these types of methods that have been proven to work in other fields of research. The merits of the RF imputation method are also proven in a more financial context in a study by Sue, Tsai and Tsau (2022). Which uses this method in the imputation of financial features such as total liability/equity ratio and earnings per share. This application is closer to that of financial features and borrower characteristics for IOM's, making it specifically relevant for this research. Especially the RF imputation method and the K-nearest-Neighbours (KNN) approach are found to work particularly well in this research. Especially in cases when higher percentages of data are missing (up to 50% in the paper by Sue et al. (2022)), the RF imputation outperforms other methods like: mean/mode, multiple imputation (MICE), Deep Neural Network (DNN) and KNN. These cases are more relevant for the case of borrower characteristics imputation, since the share of missing values are usually large. Another research that also looks into cases of large proportions of missing data is the paper by Kofman and Sharpe (2003). This study focuses on two different Multiple Imputation methods (Expectation Maximisation and Imputation Posterior) to impute the estimation of a secured status of a loan. The research finds that both these methods provide added benefit when the data used meets the assumption of Missing at Random (see Section 4), also when large proportions of the data are missing. Combined with the findings of Schenker et al. (2006), this would indicate that a Multiple Imputation approach could prove useful when imputing data for Dutch IOM's.

The uncertainty of imputations from Multiple Imputation methods, and single imputation to an equal extent, is an important factor that has for example been researched by Brand, van Buuren, le Cessie and van den Hout (2019). This research employs a variety of different combinations of imputation methods (single and multiple) and bootstrapping methods to determine the uncertainty of imputations. The research by Brand et al. (2019) finds that embedding a single imputation method within a bootstrapping percentile method leads to the most robust method of determining imputation uncertainty, avoiding statistically invalid results. Earlier research by Efron (1994) has also studied the use of three different bootstrapping techniques: Nonparametric, Full-Mechanism and Multiple Imputation bootstrap. The research finds that the use of bootstrapping can provide variance and subsequent confidence intervals estimations that are "second order" accurate, indicating that the approximation error is proportionate to $1/n^2$, hence making this method very suitable for determining the uncertainty around imputations.

3 Data

The request for the data has been sent to the client of PwC. Since it turned out not to be possible to share the requested loan level data, the decision was made to use a similar (outdated) dataset from the US. Even though this dataset does not necessarily involve interest only mortgages. The structure of the dataset and the values to be imputed are similar. Therefore, this dataset is used, and in case access is granted from a Dutch bank to a real dataset containing data on Interest Only mortages, this one will be used, employing the same methods.

This similar dataset has been sourced from the Lending Club[1]. This American online financial institution hands out peer-to-peer loans. Information on their borrowers from a few years ago has been made publicly available (shielding sensitive data from individual borrowers). The dataset consists of 850000 individual mortgage contracts in the US. Of all these contracts 75 different variables are known, consisting of borrower characteristics (like income, employment length, job title, etc.), loan characteristics (interest rate, delinquencies, outstanding principal, etc.) and more general credit information on the borrower (previous bankruptcies and delinquencies, credit balance, utilisation of maximum credit). In order to keep the dataset manageable, a random subset of 10000 observations will be used for the further research. This subset is deemed sufficient for the analysis while not being too large, hindering computations. On top of that, not all 75 variables will be included for the same reasons. Here a selection of 15 is chosen, the included variables, including descriptive statistics can be found in Table 1.

Some data transformations have been made to better represent the Dutch case and to delete outliers. The first transformation is the winsorisation of the target income variable at 400000, which is roughly in line with earlier studies such as Rauh and Shyu (2024). Another transformation that has been done is to scale the loan amount to represent a more typical Dutch mortgage dataset, for this purpose the original is multiplied by roughly 8. This number has been determined in consultation with experts on Dutch mortgage data issues.

The used substitute dataset has little to no missing values, although this raises the question whether data imputation should be neccesary, the actual (Dutch IOM) dataset will have a lot of

[1] https://www.kaggle.com/datasets/wordsforthewise/lending-club

Variable	Mean	Median	Max.	Min.	Std. Dev.	Skew.	Kurt.
Annual income	94323	80000	400000	0	60892	2.27	10.26
Collateral value	564776	465855	7244577	0	394157	3.02	22.57
DtI (%)	18.56	17.21	742.31	0	17.52	15.37	501.73
Years of experience	9.97	9.42	49.33	0	7.66	0.94	4.00
Interest rate (%)	11.6	10.7	30.8	0.6	5.7	0.20	2.93
Loan amount	149740	134496	316461	7911	82911	0.47	2.20
Mths. oldest rev. acc. open	189	171	754	5	98.15	0.92	4.08
# Mortgage accounts	2.26	2	16	0	1.73	1.25	5.89
Mths. since major derog.	771	999	999	1	406.9	-1.22	2.50
Outstanding principal	17883	15673	39663	0	10407.8	0.42	2.22
Publicly rec. defaults	0.12	0	2	0	0.33	2.40	7.10
Term length	46.03	36	60	36	11.84	0.33	1.11
Total installment balance	43078	29642	860950	0	48828.6	3.40	26.89
Total installment limit	56396	43990	919158	0	52648.7	2.54	18.43
Issue date			*Date*				
Zip code			*Dummy*				
Init. list stat. (whole/frac)			*Dummy*				

Table 1: Descriptive statistics of the selected variables, for the used subset (10000 observations) of the Lending Club data

missing values. Therefore, this makes this dataset ideal to test whether the imputation methods and following uncertainty measures are suitable, since the imputed values can be checked against the actual observations out-of-sample. For this purpose, 20% of the income observations will be deleted to mimic the Dutch case, these values will only be introduced at the evaluation step to evaluate the imputed values out-of-sample.

In order to obtain some insight in the data, the variable to be imputed, annual income, is plotted to provide insight in the distribution of observations (Figure 2). From this plot it is visible that the income in this dataset follows the typical (for income) skewed distribution with a very stretched right tail, indicating a very small number of borrowers with a very high income. Additionally, a table of some descriptive statistics of the selection of variables can also provide some insight in the data (Table 1).

Apart from univariate distributions of the variables, the correlation between different variables can also provide some insight in the properties of the used dataset. In order to visualise this, a correlogram is constructed for the 13 numerical predictors in Figure 3. Most cross correlations are modest, with a few exceptions. Firstly, the initial loan amount and current outstanding principal balance are nearly perfectly positively correlated ($\rho = 0.96$), which meets expectations. Another interesting correlation to highlight is the relatively high correlation between the number of months since the oldest revolving account was opened and the years of experience in the current job. The last interesting correlation to point out is that between the instalment (mortgage) credit limit and the outstanding instalment balance.

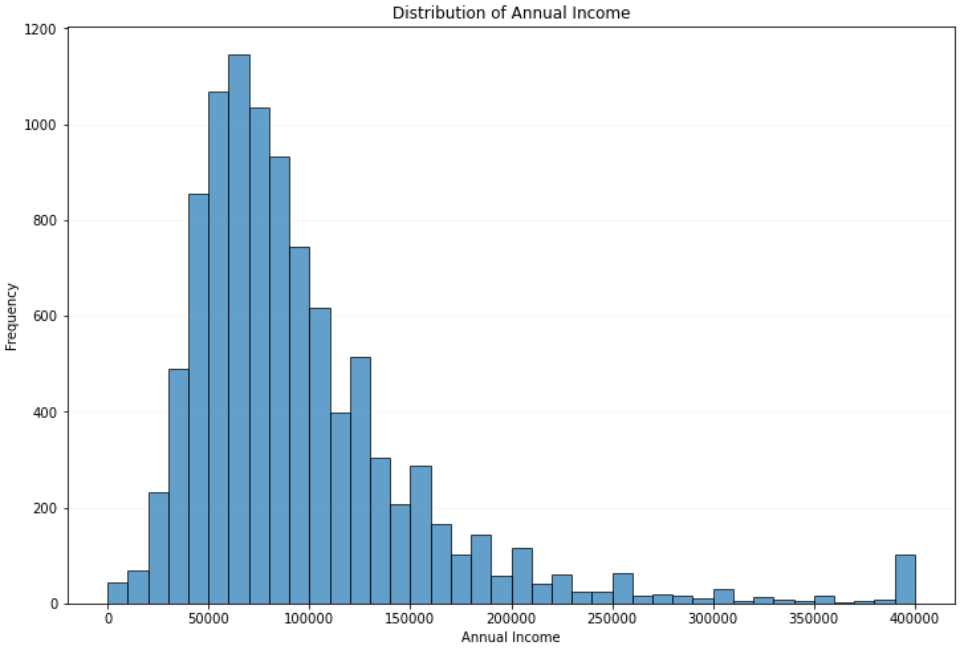

Figure 2: Income distribution, after winsorisation

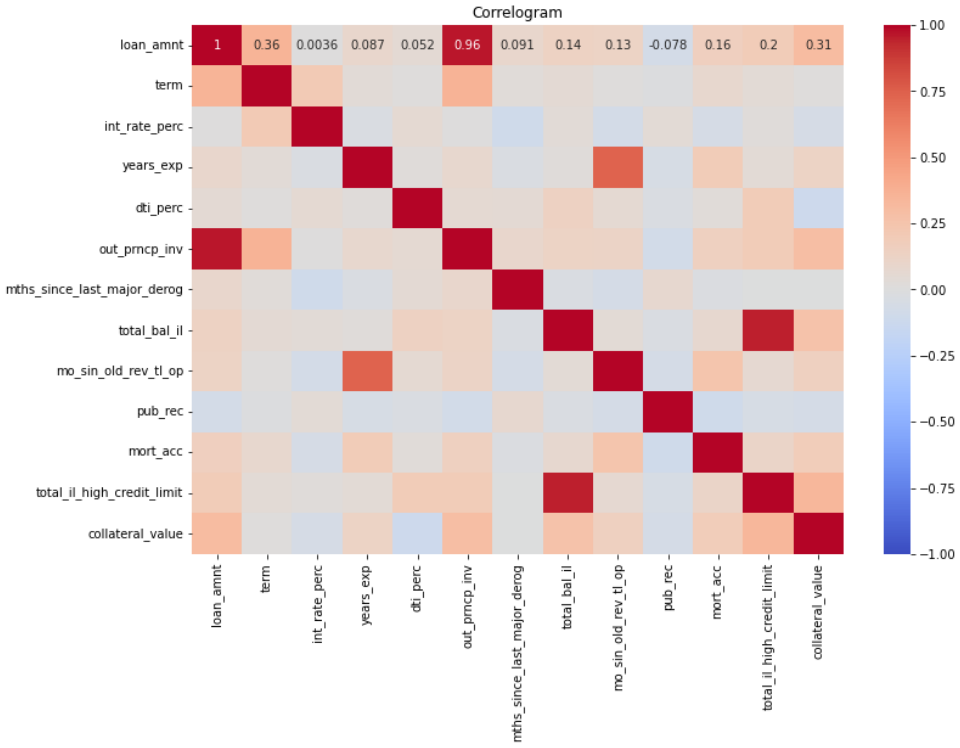

Figure 3: Correlogram for the 13 numerical predictors

4 Methodology

The methodology of this research consists of four different sections. The first one being the introduction of the methods that will be used for the imputation of the missing values in the borrower and loan characteristics. This section is divided into the four different individual imputation methods. The second part consists of the combination method to combine the indi-

vidual imputations. The third part of the methodology introduces the boostrapping approach for determining imputation uncertainty. And finally, the evaluation methods are described.

4.1 Imputation methods

A selection of imputation methods has been selected from earlier literature, where they have been shown to work well. This way, this research aims to provide multiple suitable methods, while also ensuring to have a wide spectrum of models, which is helpful when combining different imputations in the imputation combination framework.

In order to adequately impute missing values, it is important to understand the mechanism behind the missingness of the data. A widely used classification for this is introduced by Rubin (1976). This paper considers three different missing mechanisms. The first, and most simple is the case of "missing completely at random" (MCAR), which is the case when the probability of a value missing is constant over all observations. The other end of the spectrum is "missing not at random" (MNAR), in this case the probability of a value to be missing depends on both the observed data and unobserved data. In other words: there is an unobserved pattern in the data that influences whether it is missing or not. Most imputation methods, including the ones used in this research, are built on the assumption of "missing at random" (MAR). Under MAR, the assumption is that the likelihood of data being missing is unrelated to the unobserved data once the observed data is taken into account. In essence, this means that the probability of an observation missing only depends on observed data, and not on the underlying unobserved data. Mathematically, this is expressed as in Equation 1 below.

$$p(M|D) = p(M|D^{obs}), \tag{1}$$

With D being the complete dataset with both observed and unobserved data, and matrix M containing binary elements m_{ij}, which is equal to one if the data d_{ij} is observed, and zero if the data is missing.

This assumption can potentially be deemed strong, since for example in the income case, borrowers with a decreased income might be more reluctant to provide updated information to the bank out of fear of mortgage problems. However, this would only be the case if observed data is provided through voluntary updates from clients, which is not necessarily the case here.

The potentially different outcomes of the different methods can to some extent also be traced back to the assumptions that they are based on. In basis, all selected methods are based on the assumption that the data is either missing completely at random (MCAR) or missing at random (MAR). However, there are differences in how the different methods cope with a violation of these assumptions. Nonparametric models like KNN and RF seem to suffer less from MNAR, potentially because they might be able to detect underlying data patterns that follow from MNAR (Petrazzini, Naya, Lopez-Bello, Vazquez & Spangenberg, 2021). The ability of a Random Forest model in case of MNAR has been shown as well by Tang and Ishwaran (2017). However, in case the MNAR is weakly identifiable, Tseng and Chen (2019) find that regularization can also provide satisfactory performance. As opposed to the findings from Petrazzini et al. (2021), a recent study by Pereira, Abreu, Rodrigues and Figueiredo (2024) has found the KNN model

not to work well in cases of MNAR when compared to Multiple Imputation methods (in their case Multiple Imputation by Chained Equations). The research does find that in case of high proportions of data missing (up to 80%), the MI method does not perform as well in handling MNAR.

4.1.1 Ridge

The first imputation method that will be used is the Ridge regression approach, as also used in the context of borrower heterogeneity imputation by Dombrowski et al. (2022), where this method is found to work well in this context. For that reason, this method will be included in this research as well. The basis for the model that will be used is a simple regression imputation as found in Equation 2.

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon \tag{2}$$

where:

- $\hat{Y}$ is the predicted value of the dependent variable.
- β_0 is the intercept of the regression line.
- $\beta_1, \beta_2, \ldots, \beta_k$ are the regression coefficients for each independent variable $X_1, X_2, \ldots, X_k$.
- ε is the error term.

The predicted values $\hat{Y}$ are used to impute the missing values in the dataset.

But since the number of possible regressors is massive, potentially leading to multicollinearity or overfitting, a method has to be chosen to shrink the number of included regressors. One possible way to do this is by penalisation. Especially the Ridge penalisation method has been proven to work well in this context, for example in research by Dombrowski et al. (2022). This penalisation leads to a modified regression-based imputation, changing standard least squares to a penalised estimation method as can be found in Equation 3.

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{k} \beta_j^2 \right\} \tag{3}$$

where:

- $\beta_1, \beta_2, \ldots, \beta_k$ are the Ridge regression coefficients for each independent variable $X_1, X_2, \ldots, X_k$.
- λ is the Ridge penalty parameter that controls the amount of shrinkage applied to the coefficients.

The Ridge penalty parameter λ is chosen to balance the trade-off between fitting the data well and keeping the coefficients small to avoid overfitting. The predicted values $\hat{Y}$ are then used to impute the missing values in the dataset.

4.1.2 MI

Alongside the Ridge-regression based approach, a second statistical imputation method will be used in the form of Multiple Imputation as introduced by Rubin (2004) and used by Schenker et al. (2006). This methodology splits up the dataset in multiple sets in order to impute the missing values multiple times. This way, the uncertainty that comes with the imputation of data, is incorporated. The approach works by combining an Expectation Maximisation algorithm with a bootstrap framework. This bootstrapping is first used to split the dataset into multiple subsamples through resampling with replacements from the observed data. Within these subsets, basing them on both observed data and previously imputed observations, missing values are estimated by using an EM algorithm, that iteratively alternates between an Expectation and Maximisation step until convergence is reached. This process first takes the likelihood of the observed data using the law of iterated expectations as shown in Equation 4. The expected complete data log-likelihood $Q(\theta|\theta^{(i)})$ in the i-th iteration of the expectation step is taken by Equation 5. The maximisation step follows by updating the parameter estimates $\theta^{(i+1)}$ by means of maximising the previously calculated expected complete data log-likelihood $Q(\theta|\theta^{(i)})$, this step is shown in Equation 6.

$$p(D_{\text{obs}}|\theta) = \int p(D|\theta)\, dD_{\text{mis}} \tag{4}$$

$$Q(\theta|\theta^{(i)}) = E_{D_{\text{mis}}|D_{\text{obs}},\theta^{(i)}}[\log L(\theta; D_{\text{obs}}, D_{\text{mis}})] \tag{5}$$

$$\theta^{(i+1)} = \arg\max_{\theta} Q(\theta|\theta^{(i)}) \tag{6}$$

These steps are performed on each bootstrapped sample, where the parameter estimates of each of the EM algorithms of these bootstrap samples are combined. This way draws can be created from the posterior distribution of θ which is given by Equation 7. The draws from posterior of the complete-data parameters θ can then be used to draw the values of the missing observations D_{mis} from the distribution conditional on both the observed values D_{obs} and the draws of θ, by means of Equation 8.

$$p(\theta|D_{\text{obs}}) \propto p(D_{\text{obs}}|\theta) = \int p(D|\theta)\, dD_{\text{mis}} \tag{7}$$

$$D^{(m)}_{\text{mis},j} \sim p(D_{\text{mis}}|D_{\text{obs}}, \theta^{(m)}) \tag{8}$$

The use of bootstrapping adds uncertainty to the process of imputing missing values, as each iteration of the EM algorithm generates a unique arrangement of the missing data. Consequently, the final imputed dataset is constructed by combining these diverse imputed datasets, which addresses the variation that is a key aspect of any imputation process. This has the advantage of increasing efficiency of the imputed data, as also shown by Olinsky, Chen and Harlow (2003). Figure 4 below displays the Multiple Imputation steps graphically.

As research by Kofman and Sharpe (2003) has shown, Multiple Imputation methods, including one that uses an Expectation Maximisation framework as described in this Section, can

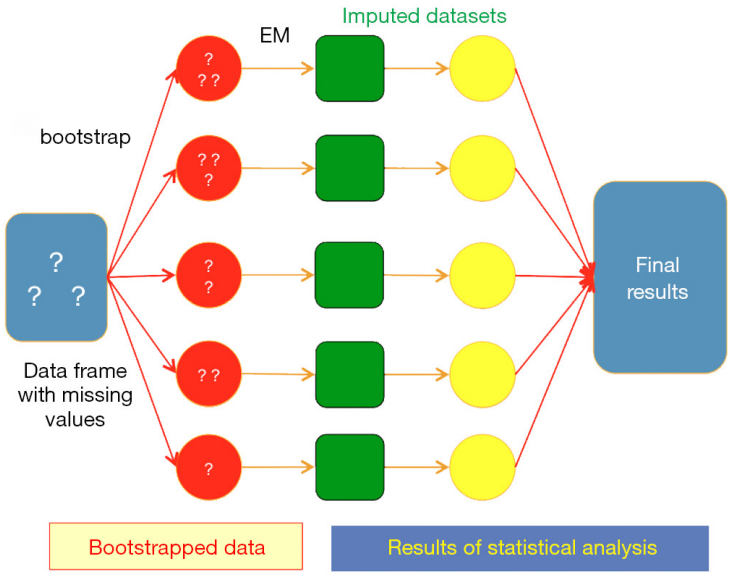

Figure 4: Graphic representation of the MI method (Zhang, 2016)

provide accurate imputations even when larger proportions of the data are missing. This property can prove very useful in the context of imputing borrower information for Dutch IOM's since the proportion of missing data is expected to be large as well. This increased accuracy is shown under the assumption of MAR, if this assumption does not hold the MI method could prove not to work as well, dependent on the proportion of data missing (Pereira et al., 2024).

4.1.3 KNN

Another widely used imputation method in this, and other contexts is the K-Nearest Neighbours (KNN) approach, as for example used by Sue et al. (2022). In this method, the missing values are imputed using values from its K nearest neighbours. A neighbour is determined by minimising a distance function, usually the Euclidian distance is chosen for this, in the form of: $d(y, z) = \sqrt{\sum_{i \in D}(x_{yi} - x_{zi})^2}$. The missing value is then simply an average of its neighbours (if utilising it for a numerical value, which is the case in the application for mortgage data in this research), this can be found in Equation 9.

$$\hat{Y} = \frac{1}{K} \sum_{i \in N_k} Y_i \tag{9}$$

where:

- $\hat{Y}$ represents the imputed value for the missing data point.
- N_k denotes the set of K nearest neighbours of the missing data point.
- Y_i signifies the observed value of the dependent variable for neighbour i.

Since the number of neighbours can become very large, especially in a mortgage dataset with a large number of borrowers, an optimal K has to be chosen in order to optimise imputation accuracy and computational efficiency. This can be done by using cross validation, to find the right balance between bias and variance. In this case the optimal K is found to be 5.

Multiple studies such as Pujianto, Wibawa, Akbar et al. (2019) and Sue et al. (2022) have shown that KNN imputation can provide accurate imputations, hence it is included in this research as well. KNN models do require the values ranges of the included features to be of similar orders of magnitude to prevent certain features from dominating due to their larger value ranges. This problem does not seem to be present for the features in the used dataset (Section 3), so features are not standardised.

4.1.4 RF

The final imputation method to be used is a supervised Machine Learning method in the form of a Random Forest (RF) imputation method. This method has been found to outperform its more traditional counterparts both in terms of imputation accuracy and narrower confidence intervals by Shah, Bartlett, Carpenter, Nicholas and Hemingway (2014) and Sue et al. (2022) among others. This ensemble learning method works by a process of iterative decision making where a large number of individual decision trees are constructed. These trees consist of a large number of individual decision rules, where each side of the decision is called a leaf. This way a decision tree partitions the predictor space into J distinct and non-overlapping regions and assigns a predicted value for each of these regions based on the mean response value in the region. This two step procedure is constructed as follows:

- Partitioning of the predictor space into J regions: $R_1, R_2, ..., R_j$.
- Assigning the predictions as the mean response value, $\bar{\pi}_{R_j}$, for a region R_j.

These regions are constructed to minimise the Residual Sum of Square (RSS) in the form of: $\text{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (\pi_i - \bar{\pi}_{R_j})^2$

Since having a very complicated model, resulting from the method described above can lead to overfitting problems, and resulting poor performance out-of-sample, bagging is used to overcome this issue. Bagging works by generating multiple decision trees from various subsets of training set and calculating an average prediction. This improves the overall stability of the model. In short, this method works in three steps:

- Generating multiple bootstrapped samples from the training set of the data.
- Building decision trees for each of these bootstrapped samples.
- Averaging the predictions of the different trees.

This technique has a downside, it can lead to highly correlated trees. In order to overcome this issue extra variation is added by only using a subset of the available predictors at each split, instead of the whole set of predictors. This way, the individual trees exhibit lower correlations between each other. This size of this subset of predictors is in this research chosen to be square root of the total number of predictors. Besides the number of predictors to include, the RF method involves a substantial amount of hyperparameters, these include: the number of trees in the forest, the number of leafs (or in other words the depth) of each tree and the minimum sample in each leaf. These hyperparameters will be optimally tuned using a grid search algorithm.

Besides more accurate predictions, also in cases of high proportions of missingness, as shown in research by Sue et al. (2022), the RF method has also been shown to better handle potential violations of the MAR assumption, for example in research by Petrazzini et al. (2021) and Tang and Ishwaran (2017). In both these researches, the RF model exhibits a smaller performance degradation when exposed to MNAR data.

4.2 Imputation combination

The merits of combinations of forecasts have been shown in literature, for example by Taylor (2020). In this paper, the imputed values from the individual methods will be combined in a similar way as that of Taylor (2020), using the Relative Score combination. In this combination, the weights of the individual imputations are inversely dependent on their individual error score. Which, in general form, calculates the weights per Equation 10 and uses those to obtain the combined forecast per Equation 11. The expectation is that, similarly as is the case for combined forecasts in for example Taylor (2020), combining individual imputations will lead to cancelling out of errors in individual imputations, thus leading to an overall more accurate imputation.

$$w_m = \frac{exp\left(-\theta \sum_{i=1}^{N} S\left(\hat{Imp}_{m,i}, y_i\right)\right)}{\sum_{j=1}^{M} exp\left(-\theta \sum_{i=1}^{N} S\left(\hat{Imp}_{j,i}, y_i\right)\right)}, \tag{10}$$

$$\hat{Imp}_{comb,i} = \sum_{m=1}^{M} w_m \hat{Imp}_{m,t}, \tag{11}$$

The parameter θ denotes the degree to which the weights are dependent on the scoring function S, in this case the Mean Absolute Error, which has been determined over a 20% in sample part of the dataset used for the weighing of the individual methods. This method uses $M = 4$ methods, in this case, over a total of $N = 10000$ observations

Subsequently, the variance of the of $\hat{Imp}_{comb,i}$ can then be calculated following an altered framework as used in case of $M = 2$ in the research by Claeskens, Magnus, Vasnev and Wang (2016) and originally by Bates and Granger (1969). This method has been altered to accompany for more individual methods, in the case of this data imputation research $M = 4$, which is shown in Equation 12.

$$\text{var}(\hat{Imp}_{comb,i}) = \sum_{m=1}^{M} w_m^2 \text{var}(\hat{Imp}_{m,t}) + \sum_{m=1}^{M} \sum_{n=1, n\neq m}^{M} w_m w_n \text{cov}(\hat{Imp}_{m,t}, \hat{Imp}_{n,t}) \tag{12}$$

Assuming that $\text{var}(\hat{Imp}_{m,t}) = \sigma_m^2$, we can write the variance of the combined prediction as:

$$\text{var}(\hat{Imp}_{comb,i}) = \sum_{m=1}^{M} w_m^2 \sigma_m^2 + \sum_{m=1}^{M} \sum_{n=1, n\neq m}^{M} w_m w_n \text{cov}(\hat{Imp}_{m,t}, \hat{Imp}_{n,t})$$

4.3 Bootstrapping

As was described in the Introduction section earlier, coming up with a way to determine a confidence interval around an imputed datapoint in order to quantify the uncertainty carried

by an imputed value, is an important aspect of data imputation which is often overlooked in other applications. A commonly used (Little & Rubin, 2019), robust way to do this is using bootstrapping as was introduced by Efron and Tibshirani (1993). This framework uses the original data to randomly construct a large number of subsets (without replacement) which can be used to make multiple calculations of the same variable of interest across these number of bootstrapped samples. The procedure is described in Equations 13, 14 and 15 below.

Given data:

$$\{X_1, X_2, \ldots, X_n\}. \tag{13}$$

Generate B bootstrap samples:

$$\{X_{1b}^*, X_{2b}^*, \ldots, X_{nb}^*\}, \text{ for } b = 1, 2, \ldots, B. \tag{14}$$

For each bootstrap sample, compute the statistic $\hat{\theta}_b^*$:

$$\hat{\theta}_b^* = f(X_{1b}^*, X_{2b}^*, \ldots, X_{nb}^*), \tag{15}$$

where $f(\cdot)$ is the function associated with the relation between the predictors (the different imputation methods) and the statistic of interest, in this case the vector of imputed income values based on the X variables in the bootstrap sample.

Since this yields a large number of imputed values for one individual observation, all based on a different bootstrapped subsample of the data. A variance for imputed value i can be constructed in the following way:

$$\hat{\text{Var}}(\hat{\theta}_i) = \frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}_i b^* - \bar{\theta_i}^*)^2, \tag{16}$$

where

$$\bar{\theta_i}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_{ib}^*.$$

Nextly, the variances resulting from the bootstrapping will be used to construct 95% confidence intervals to visualise the uncertainty corresponding to the imputed values. The assumed distribution for these intervals is a normal distribution. For this assumption, the Central Limit Theorem from (Le Cam, 1986) is used which states the following:

Formally, let $\{\theta_1, \theta_2, \ldots, \theta_n\}$ be i.i.d. random variables with mean μ and finite variance σ^2. The CLT states that as $n \to \infty$,

$$\frac{\sqrt{n}(\bar{\theta}_n - \mu)}{\sigma} \xrightarrow{d} \mathcal{N}(0, 1), \tag{17}$$

where $\bar{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \theta_i$ is the sample mean, and $\xrightarrow{d}$ denotes convergence in distribution. For the formal proof of this theorem, see Appendix A.

Resulting from this, the confidence intervals can be constructed using the mean imputed value, the bootstrap variance and the critical value $z_\alpha/2$ from the normal distribution ($\alpha = 0.05$).

$$\left[\bar{\theta_i}^* - z_{\alpha/2}\sqrt{\hat{\text{Var}}(\hat{\theta}_i)}, \bar{\theta_i}^* + z_{\alpha/2}\sqrt{\hat{\text{Var}}(\hat{\theta}_i)}\right], \tag{18}$$

4.4 Evaluation methods

The methods will be evaluated using a validation set, which is constructed by keeping 20% of the observed values out of the model estimation, to be able to treat them out-of-sample, as shown in Section 3.

The most straightforward way of evaluating individual performance of imputation methods would be to look at scoring functions such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). In this research, the MAE will be used due to the typically large magnitude of income data, as also done in Kibekbaev and Duman (2016). The exact specification of the MAE can be found in Equation 19.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{19}$$

Where:

- MAE is the Mean Absolute Error
- n is the total number of data points
- x_i is the true value of the ith data point
- $\hat{x}_i$ is the imputed (predicted) value of the ith data point

The same out-of-sample approach will be used for the second evaluation method, in the form of a back testing approach. This will be applied to the confidence intervals that are constructed using the methods described before (Section 4), to quantify and test the uncertainty that comes with imputation. Using the validation set out-of-sample, the number of violations of the confidence interval can be determined. This approach is widely used in for example Value-at-Risk forecasting and is introduced by Christoffersen (1998). This means that when constructing a 90% confidence interval, the lower bound (or upper depending on the variable) will be of 5% level (assuming symmetry). In case of income for example, the interval means that it can be said with 95% confidence that the income at maturity is above a certain level. In the validation set it is expected that roughly $\alpha = 5\%$ of observations violate the lower and upperbound of the set confidence interval combined.

The third metric, used to evaluate the constructed confidence intervals has to be viewed in conjunction with the violation rates described above. Since the construction of confidence intervals with relatively low violations can be made easy when just simply constructing a very wide confidence interval. Therefore, it helps to also look at the average confidence interval length to gain a better insight in the constructed intervals. This average confidence interval length is constructed per Equation 20 below.

$$\text{Average Length}(\text{CI}_\alpha) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\theta}_{\text{imp},i,\text{upper}} - \hat{\theta}_{\text{imp},i,\text{lower}}\right) \tag{20}$$

The final evaluation method is to check whether a selected imputation method gives values that are structurally too low or too high. Especially a consistent overestimation of the borrower's income can lead to added risks. This is specified in Equation 21 below.

$$\text{Bias}(\hat{\theta}_{\text{imp}}) = E[\hat{\theta}_{\text{imp}}] - \theta \tag{21}$$

5 Results

The results of the different methods and analyses will be presented in this section as follows: Section 5.1 will present the results from the different individual methods, Section 5.2 will show how the combined method is constructed and how it affects performance, and finally Section 5.3 will clarify some further uncertainties that might arise.

5.1 Individual methods

Firstly, the four individual methods are constructed and evaluated against the validation set that was held out-of-sample (20%). To provide some graphical insight into the performance of the individual methods, the imputed values are plotted for each of the individual methods, including their respective confidence intervals, against the true values for a subsample of 100 for ease of reading. This can be seen in Figures 5, 6, 7 and 8 for the Random Forest, KNN, MI and Ridge imputation methods respectively. From these plots it becomes apparent that the imputations for all four of the methods follow the true values reasonably well, with the grey dots being close to the actual values in orange. Interestingly, for the RF model (same for the KNN and Ridge models, for the MI model to a lesser extent) the uncertainty becomes larger (wider confidence interval) for higher incomes. This can be traced back to two things, firstly the smaller number of observations in this value range, and secondly the higher absolute differences between different observations. Simply put, there are less households making 400 thousand per year compared to 50, besides that, the difference between an income of 350 and 400 thousand is relatively smaller than that of 50 versus 100 thousand, making it more difficult to predict these values accurately.

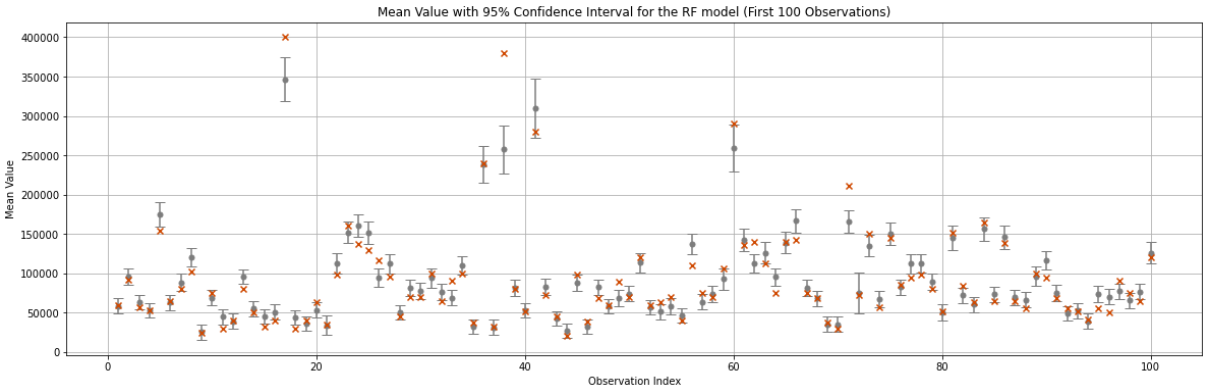

Figure 5: Plot of the imputed values and confidence intervals of the RF model

The findings from these plots are further substantiated by Table 3. From this table, it is

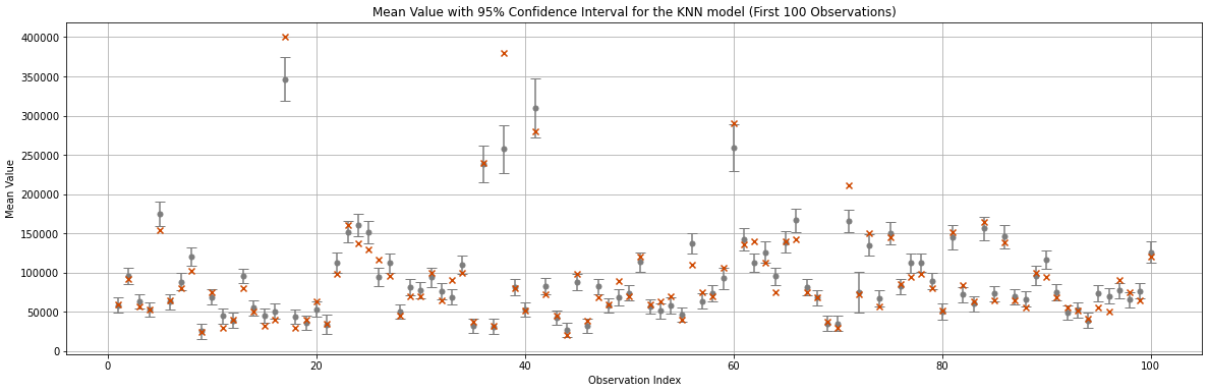

Figure 6: Plot of the imputed values and confidence intervals of the KNN model

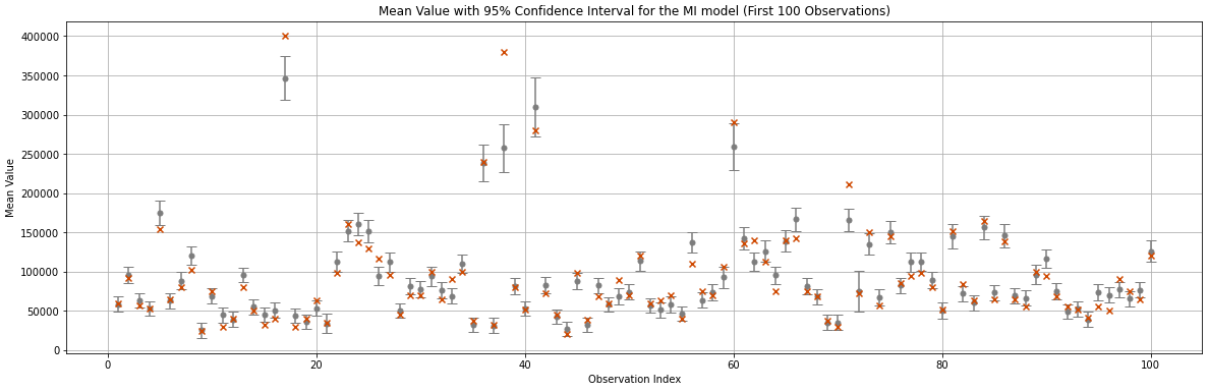

Figure 7: Plot of the imputed values and confidence intervals of the MI model

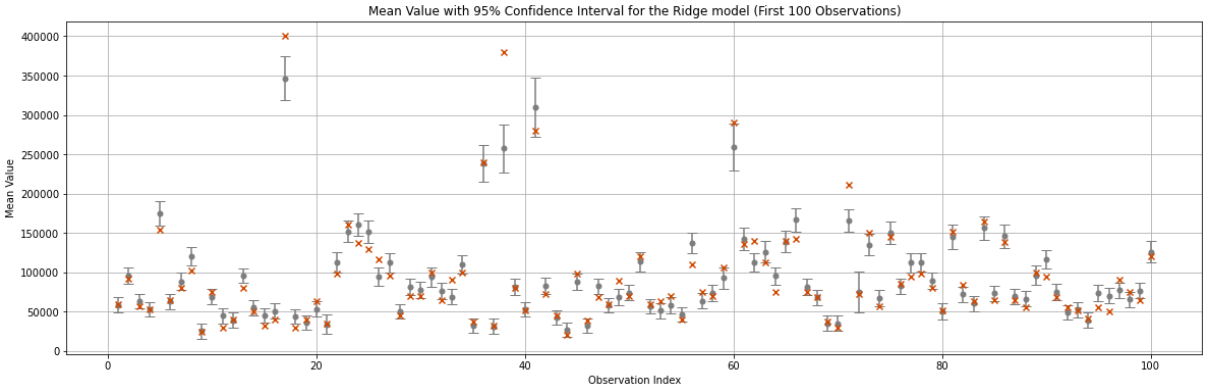

Figure 8: Plot of the imputed values and confidence intervals of the Ridge model

also visible that the RF imputation method produces the most accurate point estimates (MAE), which is a common trait of Machine Learning models, and this is in line with earlier research, for example by Sue et al. (2022) and Wang et al. (2022), that also finds the RF method to work well in an imputation context. Since the RF method has been shown in literature before to also handle MNAR cases relatively well ((Petrazzini et al., 2021), (Tang & Ishwaran, 2017)), its superior performance can also be an indication of MNAR in the data. However, since this can not be observed, this hypothesis can not be tested. In terms of MAE, the MI method seems to provide the least accurate point estimates, this further supports findings from Sue et al. (2022)

that MI methods do not perform as well as RF in cases with large proportions of data missing, as is the case in this research. This finding does seem to contradict research by Kofman and Sharpe (2003), where MI methods are found to work well when large proportions of data are missing, although it is worth noting that the aforementioned paper does not include Machine Learning methods like RF. This RF method, however, does have the highest number of violations of the confidence interval, due to the low estimated variance of these estimates, which can potentially be traced back to overfitting, which is also one of the properties of an RF model where the in-sample values are fitted very well, but the out-of-sample ones not as much.

When taking into account the other two individual methods, Ridge and KNN, it can be seen that both these methods do not provide point estimates that are as accurate as that of RF. However, both methods do manage to outperform the MI method. These findings are in line with earlier research, for example the paper by Dombrowski et al. (2022) does find a variety of Ridge based methods to work well in imputing borrower heterogeneity. This application does come close to the one from this research, which, in combination with the presented performance in Table 3, makes implementing this method in the context of income imputation for mortgages reasonable. The KNN method does produce more accurate point estimates than the Ridge based method, but it does not manage to surpass the RF method. A potential reason for this can be found in the properties of the KNN method. Since a sizeable proportion of the data is missing in this research, this can indicate that the similarity between the missing data and observed ones is small. And since the KNN method relies on this similarity, it might make this method less suitable in case a large proportion of data is missing. This effect is also shown by Emmanuel et al. (2021), where KNN was more impacted by higher missing rates than RF.

All of the estimations are negatively biased, which is a positive thing on one hand, as it means that income is underestimated, which is prudent as the opposite would mean a hidden risk for the bank. But it can also be an indicator that the data is fitted incorrectly. This is further substantiated from a poor fit of confidence intervals. Apart from the MI method, all methods show a much higher number of violations of the confidence interval than would be expected based on the level of α (5%) and the number of observations out-of-sample (2000), so 100 expected violations. This could be an indication that the combination of the estimated variance and assumed distribution is incorrect, contradicting research by Efron (1994) which does find that bootstrapping combined with imputation leads to second order accurate variance and confidence interval estimation. This number of violations is highest for the more advanced methods like RF and KNN, which can potentially be attributed to the overfitting nature of these models, leading to excellent in sample predictions, but poor out-of-sample predictions. The RF and KNN methods also exhibit the narrowest and second narrowest confidence intervals respectively. Especially the average length of the confidence intervals of the RF method are small when compared to the MI method, the ones for the MI method are eight times larger.

Another reason for the high number of observed confidence interval violations is the violation of one of the assumptions, especially the assumption of the normal distribution of the imputed values based on the Central Limit Theorem is a strong assumption and needs to be investigated further. It seems plausible that the CLT does not hold since the sample size is too small, currently 1000 bootstrapped samples are generated, but because only 10% of the observations

are used in a bootstrapped sample, it could mean that the total number of estimates (roughly $1000 * 0.1 = 100$) is too low. Therefore, the bootstrap will also be performed using a much larger 100000 iterations and checking their distribution in Section 5.3.

5.2 Combination

The individual methods can be used to construct the combined imputation as described in Section 4.2. The weights from the Relative Score combination method, as determined by Equation 10 can be found in Table 2 below. Parameter θ has been set to 0.0001.

Method	Ridge	KNN	RF	MI
Weights	24.4%	20.9%	38.2%	16.5%

Table 2: Relative Score combination weights for the four individual methods

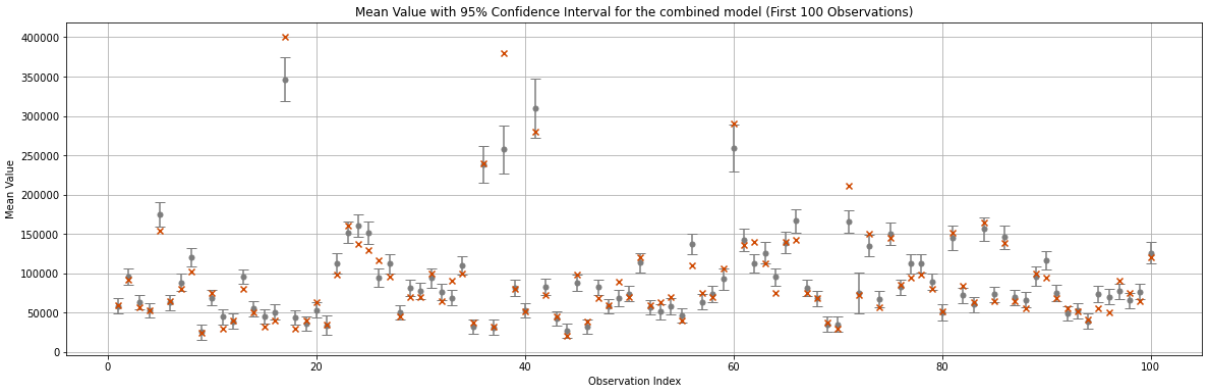

Figure 9: Plot of the imputed values and confidence intervals of the Combined model

Method	Ridge	KNN	RF	MI	Combined
MAE	13405.4	13361.7	13250.0	13876.1	13096.9
Bias	-164.542	-520.912	-665.972	-619.178	-504.760
Violations	437	777	1111	41	1210
Avg. length	34382.5	28176.1	12253.1	103167	20097.4

Table 3: Results for 1000-fold bootstrapped imputation

As can be seen in Figure 9 and Table 3, combining does yield better point estimates for the imputations. Which is in line with earlier findings from papers such as Taylor (2020), where combined forecasts (or in this research imputations) provide a more accurate outcome due to the fact that multiple methods partially cancel out each other's errors. This usually also leads to a lower variance of the estimated imputations, which is also reflected in the constructed confidence intervals for the combined imputations. These intervals are relatively narrow, when compared to other models. Which is a reason for the high number of violations of the confidence interval. As was the case with the individual methods, this points an issue in the variance estimation, this will be researched further in Section 5.3. Potentially different variance estimation and or combination methods are needed to overcome this issue.

5.3 Further investigation

As was indicated before in Sections 5.1 and 5.2, the confidence intervals do not show a number of violations that would be expected based on a 5% confidence interval and a sample size of 2000, in this case around 100 violations are expected. From Table 3 it can be seen that the true number of violations fluctuate from around 400 to more than 1200. One explanation for this is the low number of bootstrapped samples used, leading to only around 100 different estimates for each observation, which is low for the assumption of the Central Limit Theorem to hold. Therefore, the same procedure has been done, but now with 100000 bootstrapped samples, leading to approximately 10000 estimates for each observation, sufficient for the CLT. The results of this can be found in Table 4 below.

Method	Ridge	KNN	RF	MI	Combined
MAE	13374.7	13368.6	13261.2	13352.0	13053.3
Bias	-124.401	-471.405	-599.680	-399.005	-423.609
Violations	493	855	1106	42	933
Avg. length	39841.3	29078.9	23811.1	98435.9	24088.3

Table 4: Results for 100000-fold bootstrapped imputation

Unfortunately, increasing the number of bootstrapped samples does not lead to a massive improvement in the estimation of the confidence intervals. The number of violations remain roughly constant, apart from the combined imputation, which does see a decrease.

In order to gain more insight into the distribution of the individual estimates for an observation and to see if the assumption of the CLT holds that this distribution will converge to a normal distribution, the individual estimates of one random observation are plotted in Figure 10. The distribution in this plot seems to be close to that of a normal one, albeit with some slightly longer tail on the right-hand side, which together with the negatively biased point estimates could point to a violation of the normality assumptions. However, the distribution in this plot does not provide enough evidence that the CLT does not hold in this case, and therefore the reason for the poor fit of the confidence intervals could also come from something else. Potentially, the estimation of the variance is off, and different variance estimation techniques should be used. One possible method for this, as was introduced in a recent study by Knotterus (2022), is Kalman filtering, which is described in more detail in the Appendix B. Another possibility is to use a different type of bootstrapping as was also used in the context of survey non-response by Mashreghi, Léger and Haziza (2014). This paper finds that bootstrapping methods that treat the imputed values as observed, as is the case in the method used in this research, can generally lead to variance estimates that are too small (as also observed in this research). The proposed alternative method by (Mashreghi et al., 2014) makes modifications for sample surveys, independently generates the responses and imputes non-responses in the bootstrap sample. The resulting method is called an independent bootstrap and is found to provide valid variance estimations, even for large sampling fractions.

Since most of the methods carry a sizeable uncertainty in the predictions, it would be useful to look at the implication of this uncertainty in for example probability of default predictions. This will be done in the next Section 5.4.

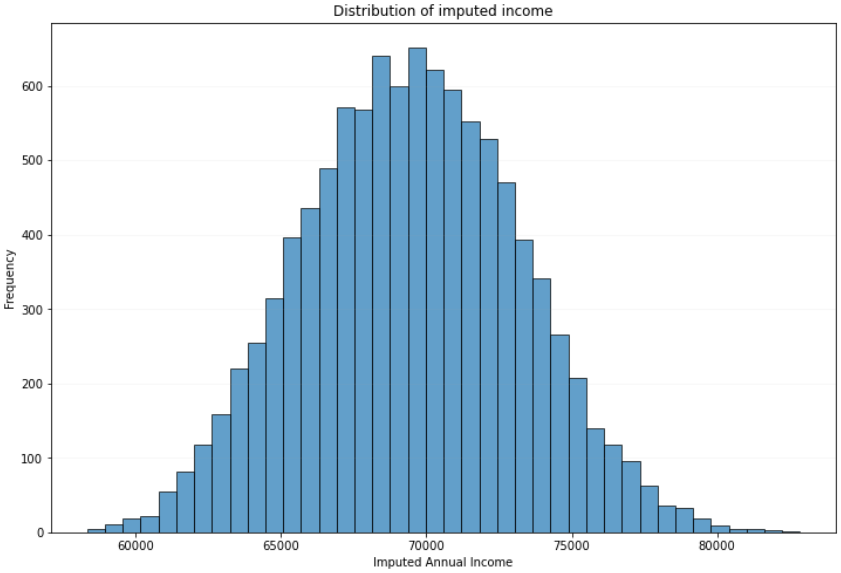

Figure 10: Distribution of the imputed values for one observation

5.4 Implication of uncertainty

In order to quantify the implication of the uncertainty, carried by the imputation of missing value, the impact of this uncertainty on probabilities of default is evaluated. For this purpose, a dataset is needed which contains a default indicator to train and evaluate the PD model. This indicator is absent in the dataset used earlier in this research. Therefore, a different one is sourced from the American Federal National Mortgage Association, commonly known as Fannie Mae [2]. This dataset includes a large number of variables, among which the Debt-To-Income (DTI) ratio and a variable that indicates the delinquency of a loan. The full list of the variables that are in the dataset, after cleaning largely empty columns can be found in Appendix C. From this, cleaned dataset, a 100 thousand subsample is drawn, which is split in a 70% training and 30% testing set. The training set is used to train a logistic regression default prediction model, as first introduced by Collins (1980) and Ohlson (1980). This method is still widely used and is considered the benchmark model in default predictions because of both interpretability and ease of use (Westgaard & Van der Wijst, 2001). The estimation of this model can be found in Equation 22 and 23

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \sum_{j=1}^{J} \beta_j x_{ji} = \boldsymbol{\beta}\boldsymbol{x}_i, \quad \forall i = 1, \ldots, N, \tag{22}$$

with $p_i = P(y_i = 1|X = \boldsymbol{x}_i)$, the conditional probability of default for observation i, and $\boldsymbol{x}_i$, the corresponding J covariates. Element j in the parameter vector $\boldsymbol{\beta}$ resembles the change in the log-odds ratio of default for a change in the predictor j (ceteris paribus). To obtain the probability of default for observation i, Equation 22 can be rewritten to Equation 23 below.

$$p_i(\boldsymbol{x}_i; \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}\boldsymbol{x}_i)}{1 + \exp(\boldsymbol{\beta}\boldsymbol{x}_i)} = \frac{1}{1 + \exp(-\boldsymbol{\beta}\boldsymbol{x}_i)}. \tag{23}$$

[2] https://capitalmarkets.fanniemae.com/credit-risk-transfer/single-family-credit-risk-transfer/fannie-mae-single-family-loan-performance-data

Using maximum likelihood estimation (MLE) to estimate $\boldsymbol{\beta}$.

Since the class of defaults in the dataset is a lot smaller than that of non-defaults, the classes are highly imbalanced. As imbalanced classes would lead to an overestimation of the probability of predicting the majority class, oversampling is needed to overcome this issue (Zheng, Cai & Li, 2015). To determine the optimal threshold for which a prediction qualifies as default (1) is determined by maximising the area under the Receiver Operating Characteristic (ROC) curve, as was described in the research of Fawcett (2006). This was determined to be 0.594 in this case.

The same 70% training set from the training of the default model will be used in the same bootstrap framework as described in Section 4 to impute the DTI ratio variable of the remaining 30% of observations that are in the testing set, without including these known values to keep them out-of-sample. The chosen imputation method is the Random Forest method, as this method proved to provide the best point estimates in Section 5. The same methods as before have been used for determining the variance and corresponding confidence intervals for these points estimates.

The Final step in determining the implication of the uncertainty of the imputations is to use the trained default prediction model on three different testing sets, firstly the true out-of-sample values, secondly the imputed values of this testing set, and finally the most important one, the upper bound of the constructed 5% confidence intervals. The bound chosen here is the upper bound, since this dataset has the DTI ratio as the variable that indicates the income of the borrower. And in order to take the "worst case" (lowest) scenario for the income of the borrower, one should look at the highest end of the DTI ratio spectrum, as this ratio is inversely related to the borrower's income.

Table 5 below shows the average predicted probabilities of default among the 30% out-of-sample set, since the classifications are binary, this means the number of defaults divided by the total number of observations (roughly 30000).

Used values	True	Imputed (p-value)	5% confidence interval (p-value)
Average PD	0.2395	0.2391 (0.15)	0.2421 (< 0.01)

Table 5: Average predicted PD for the different out-of-sample values of DTI

From this table it becomes apparent that, as expected, the average predicted PD is slightly higher when using the 5% upper bound, indicating that a higher DTI ratio leads to a slightly higher probability of default for a loan. Although the difference between using the true values and the 5% confidence level is significant, it has to be noted that this difference is small (a little over a 1% relative difference in average predicted PD) and will not have large impacts on for example the provisions held by the bank. Therefore, it can be concluded that the implication of this uncertainty that is carried by the imputed values is in this case relatively small. This further accentuates the merits of data imputation in this context.

6 Conclusion

This research aims to assess the suitability and relative performance of multiple individual imputation methods and one combined method, in the context of imputing borrower characteristics for Dutch Interest Only Mortgages. Besides this, the implication of the inevitable imputation uncertainty is also a focus of the research. This is done by imputing income data using existing data in a loan level dataset containing multiple borrower characteristics. The methods used range from statistical methods in the form of Multiple Imputation, to a Ridge regression as regularisation technique and finally two supervised Machine Learning approaches, K-Nearest Neighbours and Random Forest.

The research finds that point predictions of the RF method are most accurate out of all the individual methods. This is in line with earlier literature and can be attributed to the flexible nature of this model. The RF method is followed by that of the KNN model and the Ridge models respectively, these models have been shown to work well in literature, and this research supports that finding. The MI method provides the least accurate point estimates, although it is worth noting that this model does benefit from increasing the number of bootstrapped samples, in the case of 100000 bootstrapped samples it does outperform both the KNN and Ridge models.

One of the most important findings of this research is that combining multiple different imputations will lead to a more accurate overall imputation. This has been shown for forecasts before in literature, but not in an imputation context. The method that is used for determining the weights for the combined imputation is a Relative Score combination.

The second part of the research focuses on the uncertainty carried by imputed values. The research finds large differences between the estimated variances and corresponding confidence intervals for the different methods. Both the supervised machine learning methods by the likes of RF and KNN provide the lowest variance and subsequently the narrowest confidence intervals. This is followed by the Ridge imputation and finally, the MI method provides substantially more variance and larger confidence intervals. These wider intervals for the MI model lead to a lower number of violations, when evaluating the intervals out-of-sample, this number is under the expected number based on the number of observations and chosen confidence level. The other three models exhibit substantially higher numbers of violations of the confidence intervals, higher than expected from the number of observations and confidence level. This does raise a problem as it indicates an incorrect specification of either the variance of the imputations or the assumed distribution. This has been further investigated and has been found not to be related to the used number of bootstrapped samples. A potential reason that has been found is a fatter right tail in the distribution of the income imputations, this could be a reason for the normality assumption of the CLT to not hold in this case.

The last part of this research focuses on the implication of the uncertainty that is inevitably carried by imputation of values. The research finds that the impact of the uncertainty around the point estimates does have a significant effect on average predicted probabilities of default of loans, but this effect is in fact small. Therefore, the uncertainty as calculated in this research does not have a large implication on PD modelling, making imputation a useful tool in this case.

In short, this research aims to answer the research question:

What is the impact of various imputation methods on assessing Interest Only Mortgage risk, and how can imputation uncertainty be quantified and integrated into risk modelling frameworks?

The answer to this question consists of four parts, firstly it becomes apparent that the flexible nature of an RF model makes this method especially suitable for imputing borrower characteristics. However, its performance can be improved further by combining it with other methods, to cancel out errors in individual methods. Nextly, it can be concluded that the chosen bootstrapping method of variance estimation is only limitedly suitable for quantifying the imputation uncertainty. The uncertainty, as calculated in its current form does not have a sizeable impact om risk modelling.

It is important to note a few shortcomings of this research that can be overcome in further research. Firstly, this research uses an American dataset on residential loans, which is only limitedly representative of the Dutch IOM market. With access to a Dutch IOM loan level dataset, this problem can be solved. A second important shortcoming of this research is the chosen method for estimating imputation variance, this method has been shown to provide incorrectly specified confidence intervals for three of the four methods. In further research, this can be investigated by looking at other possible methods for variance estimation such as Kalman Filtering (Knotterus, 2022) or an independent bootstrapping approach (Mashreghi et al., 2014).

References

Aron, J. & Muellbauer, J. (2016). "modelling and forecasting mortgage delinquency and foreclosure in the uk.". *Journal of Urban Economics*, *94*, 32-53. Retrieved from https://www.sciencedirect.com/science/article/pii/S0094119016000255 doi: https://doi.org/10.1016/j.jue.2016.03.005

Autoriteit Financiële Markten (AFM). (2023, August). *Bijna twee derde minder klanten met verhoogd risico bij aflossingsvrije hypotheek.* Retrieved from

Bates, J. M. & Granger, C. W. (1969). The combination of forecasts. *Journal of the operational research society*, *20*(4), 451–468.

Brand, J., van Buuren, S., le Cessie, S. & van den Hout, W. (2019). Combining multiple imputation and bootstrap in the analysis of cost-effectiveness trial data. *Statistics in Medicine*, *38*(2), 210–220.

Burrows, R. (1998). Mortgage indebtedness in england: an'epidemiology'. *Housing Studies*, *13*(1), 5–21.

Campbell, J. Y. & Cocco, J. F. (2015). A model of mortgage default. *The Journal of Finance*, *70*(4), 1495–1554.

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International economic review*, 841–862.

Claeskens, G., Magnus, J. R., Vasnev, A. L. & Wang, W. (2016). The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, *32*(3), 754–762.

Collins, R. A. (1980). An empirical comparison of bankruptcy prediction models. *Financial Management*, 52–57.

de Haan, L. & Mastrogiacomo, M. (2020). Loan to value caps and government-backed mortgage insurance: Loan-level evidence from dutch residential mortgages. *De Economist*, *168*(4), 453–473.

Dombrowski, T., Pace, R. K. & Wang, J. (2022). Imputing borrower heterogeneity and dynamics in mortgage default models. *The Journal of Real Estate Finance and Economics*, 1–26.

Efron, B. (1994). Missing data, imputation, and the bootstrap. *Journal of the American Statistical Association*, *89*(426), 463–475.

Efron, B. & Tibshirani, R. J. (1993). An introduction to the bootstrap chapman & hall. *New York*, *436*.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B. & Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big data*, *8*, 1–37.

European Systemic Risk Board. (2022). *Assessment of the dutch notification in accordance with article 458 of regulation (eu) no 575/2013 concerning application of a stricter national measure for residential mortgage lending.* Retrieved from https://www.esrb.europa.eu/pub/pdf/other/esrb.opinion220906$_r$*eport* 6*ab*688952*a.en.pdf* (Accessed: 2024-04-05)

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, *27*(8), 861–874.

Jackson, J. R. & Kaserman, D. L. (1980). Default risk on home mortgage loans: a test of competing hypotheses. *Journal of risk and insurance*, 678–690.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Kibekbaev, A. & Duman, E. (2016). Benchmarking regression algorithms for income prediction modeling. *Information Systems*, *61*, 40–52.

Knotterus, P. (2022). *On kalman filtering, parameter uncertainty and variances after single and multiple imputation*. Statistics Netherlands.

Kofman, P. & Sharpe, I. G. (2003). Using multiple imputation in the analysis of incomplete observations in finance. *Journal of Financial Econometrics*, *1*(2), 216–249.

Lambrecht, B., Perraudin, W. & Satchell, S. (1997). Time to default in the uk mortgage market. *Economic Modelling*, *14*(4), 485–499.

Le Cam, L. (1986). The central limit theorem around 1935. *Statistical science*, 78–91.

Little, R. J. & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

Mashreghi, Z., Léger, C. & Haziza, D. (2014). Bootstrap methods for imputed data from regression, ratio and hot-deck imputation. *Canadian Journal of Statistics*, *42*(1), 142–167.

Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research*, 109–131.

Olinsky, A., Chen, S. & Harlow, L. (2003). The comparative efficacy of imputation methods for missing data in structural equation modeling. *European Journal of Operational Research*, *151*(1), 53–79.

Pereira, R. C., Abreu, P. H., Rodrigues, P. P. & Figueiredo, M. A. (2024). Imputation of data missing not at random: Artificial generation and benchmark analysis. *Expert Systems with Applications*, *249*, 123654.

Petrazzini, B. O., Naya, H., Lopez-Bello, F., Vazquez, G. & Spangenberg, L. (2021). Evaluation of different approaches for missing data imputation on features associated to genomic data. *BioData mining*, *14*, 1–13.

Pujianto, U., Wibawa, A. P., Akbar, M. I. et al. (2019). K-nearest neighbor (k-nn) based missing data imputation. In *2019 5th international conference on science in information technology (icsitech)* (pp. 83–88).

Rauh, J. & Shyu, R. (2024). Behavioral responses to state income taxation of high earners: evidence from california. *American Economic Journal: Economic Policy*, *16*(1), 34–86.

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

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys* (Vol. 81). John Wiley & Sons.

Schenker, N., Raghunathan, T. E., Chiu, P.-L., Makuc, D. M., Zhang, G. & Cohen, A. J. (2006). Multiple imputation of missing income data in the national health interview survey. *Journal of the American Statistical Association*, *101*(475), 924–933.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O. & Hemingway, H. (2014). Comparison of random forest and parametric imputation models for imputing missing data using mice: a caliber study. *American journal of epidemiology*, *179*(6), 764–774.

Sue, K.-L., Tsai, C.-F. & Tsau, H.-M. (2022). Missing value imputation and the effect of feature normalisation on financial distress prediction. *Journal of Experimental & Theoretical*

Artificial Intelligence, 1–17.

Sylvain Broyer, B. S. G. A. G., Marion Amiot. (2023, February). *European housing prices: A sticky, gradual decline.* Retrieved from https://www.suerf.org/wp-content/uploads/2023/12/f$_5$3025a28d2d0872c428da719fa518a46$_6$0823$_s$

Tang, F. & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *10*(6), 363–377.

Taylor, J. W. (2020). Forecast combinations for value at risk and expected shortfall. *International Journal of Forecasting*, *36*(2), 428–441.

Tseng, C.-h. & Chen, Y.-H. (2019). Regularized approach for data missing not at random. *Statistical methods in medical research*, *28*(1), 134–150.

Wang, H., Tang, J., Wu, M., Wang, X. & Zhang, T. (2022). Application of machine learning missing data imputation techniques in clinical decision making: taking the discharge assessment of patients with spontaneous supratentorial intracerebral hemorrhage as an example. *BMC Medical Informatics and Decision Making*, *22*, 1–14.

Westgaard, S. & Van der Wijst, N. (2001). Default probabilities in a corporate bank portfolio: A logistic model approach. *European journal of operational research*, *135*(2), 338–349.

Wetten.nl-Regeling-Faillisementswet. (2024, Jan). *Faillissementswet.* Retrieved from https://wetten.overheid.nl/BWBR0001860/2024-01-01

Zhang, Z. (2016). Multiple imputation for time series data with amelia package. *Annals of translational medicine*, *4*(3).

Zheng, Z., Cai, Y. & Li, Y. (2015). Oversampling method for imbalanced classification. *Computing and Informatics*, *34*(5), 1017–1037.

A CLT for confidence intervals

Theorem (Central Limit Theorem). Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed (i.i.d.) random variables with mean μ and variance σ^2. The sample mean is defined as $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then, as $n \to \infty$,

$$\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{d} \mathcal{N}(0,1), \tag{24}$$

Proof.
Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with $E[X_i] = \mu$ and $\mathrm{Var}(X_i) = \sigma^2$. Define the standardized sum S_n as follows:

$$S_n = \frac{1}{\sigma\sqrt{n}} \sum_{i=1}^{n} (X_i - \mu). \tag{25}$$

To prove the Central Limit Theorem, we will show that $S_n \xrightarrow{d} \mathcal{N}(0,1)$ as $n \to \infty$.
First, consider the moment generating function (MGF) of X_i. The MGF of a random variable X is defined as $M_X(t) = E[e^{tX}]$. For the standardized variable $(X_i - \mu)/\sigma$, the MGF is given by:

$$M_{(X_i-\mu)/\sigma}(t) = E\left[e^{t(X_i-\mu)/\sigma}\right]. \tag{26}$$

Since X_i are i.i.d., the MGF of the sum $\sum_{i=1}^{n}(X_i - \mu)$ is:

$$M_{\sum_{i=1}^{n}(X_i-\mu)}(t) = \left(M_{X_i-\mu}(t)\right)^n. \tag{27}$$

By the properties of MGFs and considering the linear transformation, the MGF of S_n is:

$$M_{S_n}(t) = E\left[e^{tS_n}\right] = E\left[e^{\frac{t}{\sigma\sqrt{n}}\sum_{i=1}^{n}(X_i-\mu)}\right] = \left(M_{X_i-\mu}\left(\frac{t}{\sigma\sqrt{n}}\right)\right)^n. \tag{28}$$

For small t, we can expand $M_{X_i-\mu}\left(\frac{t}{\sigma\sqrt{n}}\right)$ using a Taylor series around $t = 0$:

$$M_{X_i-\mu}\left(\frac{t}{\sigma\sqrt{n}}\right) \approx 1 + \frac{t}{\sigma\sqrt{n}}E[X_i - \mu] + \frac{1}{2}\left(\frac{t}{\sigma\sqrt{n}}\right)^2 E[(X_i-\mu)^2]. \tag{29}$$

Since $E[X_i - \mu] = 0$ and $E[(X_i - \mu)^2] = \sigma^2$, this reduces to:

$$M_{X_i-\mu}\left(\frac{t}{\sigma\sqrt{n}}\right) \approx 1 + \frac{t^2\sigma^2}{2n\sigma^2} = 1 + \frac{t^2}{2n}. \tag{30}$$

Taking the nth power, we get:

$$\left(1 + \frac{t^2}{2n}\right)^n \approx \exp\left(\frac{t^2}{2}\right). \tag{31}$$

Therefore,

$$M_{S_n}(t) \approx \exp\left(\frac{t^2}{2}\right), \tag{32}$$

which is the MGF of the standard normal distribution $\mathcal{N}(0,1)$. Hence, by the uniqueness theorem for MGFs, $S_n \xrightarrow{d} \mathcal{N}(0,1)$.
Proving the Central Limit Theorem.

B Kalman filtering

As was described in the Introduction section earlier, coming up with a way to determine a confidence interval around an imputed datapoint in order to quantify the uncertainty carried by an imputed value, is an important aspect of data imputation which is often overlooked in other applications. A common way to do this is using bootstrapping, however, recent research by Dutch bureau of statistic, CBS (Knotterus, 2022), has introduced a novel approach to determine these confidence intervals using Kalman Filtering (Kalman, 1960) and the corresponding update equations.
In these Kalman updating equations, as can be found in Equations 33, 34 and 35, the parameters b of the full sample n are updated, from the initial estimation based on the observed sample p, to the estimation including the imputed values q.

$$b_n = b_{p'} + K_q e_q \tag{33}$$

$$V_n = I_k - K_q X_q V_p \tag{34}$$

$$K_q \equiv V_p X_q' (X_q V_p X_q' + F_q^{-1})^{-1} \tag{35}$$

Where:

- $\mathbf{F}_l$: Diagonal matrix with elements f_i, the known weights associated with the errors in the linear model
- $\mathbf{V}_l$: Covariance matrix of errors.
- $\mathbf{K}_q$: Kalman gain.

This does then yield the prediction error for imputed value y_q given y_p as $e_q = X_q v + u_q$. With u_i being the random error, which is assumed to be normally distributed. Using this framework, Knotterus (2022) have shown to be able to construct prediction (or imputation) errors of imputed values, which can then consequently be used to calculate the confidence intervals for the imputed values.

C Included variables in the Fanny Mae dataset

Variable	Name
1	Loan Identifier
2	Monthly Reporting Period
3	Channel
4	Seller Name
5	Servicer Name
6	Original Interest Rate
7	Current Interest Rate
8	Original UPB
9	Current Actual UPB
10	Original Loan Term
11	Origination Date
12	First Payment Date
13	Loan Age
14	Remaining Months to Legal Maturity
15	Remaining Months To Maturity
16	Maturity Date
17	Original Loan to Value Ratio (LTV)
18	Original Combined Loan to Value Ratio (CLTV)
19	Number of Borrowers
20	Debt-To-Income (DTI)
21	Borrower Credit Score at Origination
22	First Time Home Buyer Indicator
23	Loan Purpose
24	Property Type
25	Number of Units
26	Occupancy Status
27	Property State
28	Metropolitan Statistical Area (MSA)
29	Zip Code Short
30	Current Loan Delinquency Status
31	Servicing Activity Indicator
32	Relocation Mortgage Indicator
33	High Balance Loan Indicator

D Programming code

-*- coding: utf-8 -*- """ Created on Wed Jul 3 10:45:21 2024

@author: wkoomen001 """

import numpy as np import pandas as pd from sklearn.linear$_{m}odelimportRidgefromsklearn.neighborsimportKNeighborsRegressorfromsklearn.ense$

Load the data df = pd.read$_{e}xcel($"$C:$

$Users$

$wkoomen$001

$Documents$

$thesis$

$Kaggle$

$Missing_{s}ubsample_{1}0000_{n}ew_{5}var_{C}hanged.xlsx")df = pd.read_{e}xcel("C:$

$Users$

$wkoomen$001

$Documents$

$thesis$

$Kaggle$

$Missing_{s}ubsample_{1}0000_{n}ew_{1}5var_{c}hanged.xlsx")Fornewdatadrop_{c}olumns1 = [' member_{i}d']df = df.drop(drop_{c}olumns1, axis = 1)$

factorise non numerical columns data$_{e}ncoded = pd.get_{d}ummies(df, columns = [' zip_{c}ode',' initial_{l}ist_{s}tatus'])drop_{c}olumns3 = [' issue'_{d}]df = data_{e}ncoded.drop(drop_{c}olumns3, axis = 1)$

def bootstrap$_{i}mputed_{m}atrix_{c}ombined(data, variable_{n}ame, m, alpha = 1.0, n_{n}eighbors = 5) : Placeholderforimputedmatricesimputed_{m}atrix_{r}idge = np.full((len(data), m), np.nan)imputed_{m}atrix_{k}nn = np.full((len(data), m), np.nan)imputed_{m}atrix_{r}f = np.full((len(data), m), np.nan)sample_{s}ize = 1000Samplesizeforbootstrap$

for imp in range(m): Bootstrap resampling bootstrap$_{d}ata = np.random.choice(data.index, size = sample_{s}ize, replace = False)imp_{d}ata = data.loc[bootstrap_{d}ata].copy()$

Separate data with and without missing values missing$_{r}ows = np.isnan(imp_{d}ata[variable_{n}ame])X_{c}omplete = imp_{d}ata[\ missing_{r}ows].drop(variable_{n}ame, axis = 1)Y_{c}omplete = imp_{d}ata[\ missing_{r}ows][variable_{n}ame]X_{m}issing = imp_{d}ata[missing_{r}ows].drop(variable_{n}ame, axis = 1)$

Ridge Regression imputation ridge$_{m}odel = Ridge(alpha = alpha)ridge_{m}odel.fit(X_{c}omplete, Y_{c}omplete)y_{m}issing_{p}redicted_{r}idge = ridge_{m}odel.predict(X_{m}issing)imputed_{m}atrix_{r}idge[bootstrap_{d}ata[missing_{r}ows], imp] = y_{m}issing_{p}redicted_{r}idgey_{c}omplete_{p}redicted_{r}idge = ridge_{m}odel.predict(X_{c}omplete)imputed_{m}atrix_{r}idge[bootstr$

$y_{c}omplete_{p}redicted_{r}idge$

KNN imputation knn$_{m}odel = KNeighborsRegressor(n_{n}eighbors = n_{n}eighbors)knn_{m}odel.fit(X_{c}omplete, Y_{c}omplete)y_{m}issing_{p}redicted_{k}nn = knn_{m}odel.predict(X_{m}issing)imputed_{m}atrix_{k}nn[bootstrap_{d}ata[missing_{r}ows], imp] = y_{m}issing_{p}redicted_{k}nny_{c}omplete_{p}redicted_{k}nn = knn_{m}odel.predict(X_{c}omplete)imputed_{m}atrix_{k}nn[bootstrap_{d}at$

$y_{c}omplete_{p}redicted_{k}nn$

Random Forest imputation $\text{rf}_{m}odel = RandomForestRegressor()rf_{m}odel.fit(X_{c}omplete, Y_{c}omplete)y_{m}issing_{p}$
$rf_{m}odel.predict(X_{m}issing)imputed_{m}atrix_{r}f[bootstrap_{d}ata[missing_{r}ows], imp] =$
$y_{m}issing_{p}redicted_{r}fy_{c}omplete_{p}redicted_{r}f = rf_{m}odel.predict(X_{c}omplete)imputed_{m}atrix_{r}f[bootstrap_{d}ata[\ mis$
$y_{c}omplete_{p}redicted_{r}f$

return $\text{imputed}_{m}atrix_{r}idge, imputed_{m}atrix_{k}nn, imputed_{m}atrix_{r}f$

Example usage $\text{imputed}_{m}atrix_{r}idge, imputed_{m}atrix_{k}nn, imputed_{m}atrix_{r}f =$
$bootstrap_{i}mputed_{m}atrix_{c}ombined(df, "annual_{i}nc_{w}ins", m = 100000)$

Compute observation means for each method $\text{observation}_{m}eans_{r}idge =$
$np.nanmean(imputed_{m}atrix_{r}idge, axis = 1)observation_{m}eans_{k}nn =$
$np.nanmean(imputed_{m}atrix_{k}nn, axis = 1)observation_{m}eans_{r}f =$
$np.nanmean(imputed_{m}atrix_{r}f, axis = 1)$

Compute observation variances for each method $\text{observation}_{v}ariances_{r}idge =$
$np.nanvar(imputed_{m}atrix_{r}idge, axis = 1)observation_{v}ariances_{k}nn =$
$np.nanvar(imputed_{m}atrix_{k}nn, axis = 1)observation_{v}ariances_{r}f =$
$np.nanvar(imputed_{m}atrix_{r}f, axis = 1)$

Compute standard deviations for each method $\text{st}_{d}ev_{r}idge =$
$np.sqrt(observation_{v}ariances_{r}idge)st_{d}ev_{k}nn = np.sqrt(observation_{v}ariances_{k}nn)st_{d}ev_{r}f =$
$np.sqrt(observation_{v}ariances_{r}f)$

Calculate confidence intervals using normal distribution z = 2 Z-value for 95

$\text{margin}_{o}f_{e}rror = z * st_{d}ev_{r}idgeMarginoferrorlower_{b}ound =$
$observation_{m}eans_{r}idge - margin_{o}f_{e}rorupper_{b}ound = observation_{m}eans_{r}idge +$
$margin_{o}f_{e}rorCreateaDataFramefortheconfidenceintervalsconfidence_{i}ntervals_{n}ormal_{r}idge =$
$pd.DataFrame('lower_{b}ound' : lower_{b}ound,' upper_{b}ound' : upper_{b}ound, index = df.index)$

$\text{margin}_{o}f_{e}rror = z * st_{d}ev_{k}nnMarginoferrorlower_{b}ound =$
$observation_{m}eans_{k}nn - margin_{o}f_{e}rorupper_{b}ound = observation_{m}eans_{k}nn +$
$margin_{o}f_{e}rorCreateaDataFramefortheconfidenceintervalsconfidence_{i}ntervals_{n}ormal_{k}nn =$
$pd.DataFrame('lower_{b}ound' : lower_{b}ound,' upper_{b}ound' : upper_{b}ound, index = df.index)$

$\text{margin}_{o}f_{e}rror = z * st_{d}ev_{r}fMarginoferrorlower_{b}ound =$
$observation_{m}eans_{r}f - margin_{o}f_{e}rorupper_{b}ound = observation_{m}eans_{r}f +$
$margin_{o}f_{e}rorCreateaDataFramefortheconfidenceintervalsconfidence_{i}ntervals_{n}ormal_{r}f =$
$pd.DataFrame('lower_{b}ound' : lower_{b}ound,' upper_{b}ound' : upper_{b}ound, index = df.index)$

Introduce Observed values, to calculate the mse's Load the true data full = $\text{pd.read}_{e}xcel("C :$
$Users$
$wkoomen001$
$Documents$
$thesis$
$Kaggle$
$Full_{s}ubsample_{1}0000_{n}ew_{5}var_{C}hanged.xlsx")observed_{v}alues = full['annual_{i}nc_{w}ins']$

Calculate in sample mse's

def $\text{calculate}_{i}nsample_{m}se(actual, predicted) : actual = np.array(actual)predicted =$
$np.array(predicted)onlyfirst2000, insampleactual = actual[2000 : 4000]predicted =$

$predicted[2000 : 4000]SAE = 0foriiinrange(len(predicted)) : SAE+ = np.abs(actual[i] - predicted[i])MAE = SAE/len(predicted)$
mse = np.mean((actual-predicted) ** 2)
return MAE
$\text{mse}_ridge = calculate_insample_mse(observed_values, observation_means_ridge)mse_knn = calculate_insample_mse(observed_values, observation_means_knn)mse_rf = calculate_insample_mse(observed_values, observation_means_rf)mse_mi = calculate_insample_mse(observed_values, observation_means_mi)mse_combined = calculate_insample_mse(observed_values, observation_means_combined)$
as csv with open('C:
Users
wkoomen001
Documents
thesis
$\text{imputed}_matrix_mi_100k.csv', mode =' r')asfile : csvFile = csv.reader(file)data = list(csvFile)$
data = [row for row in data[1:]] $\text{data}_array = np.array(data, dtype = float)ascsvwithopen('C :$
$Users$
$wkoomen001$
$Documents$
$thesis$
$imputed_matrix_mi_100k_2.csv', mode =' r')asfile : csvFile = csv.reader(file)data = list(csvFile)$
data = [row for row in data[1:]] $\text{data}_array_2 = np.array(data, dtype = float)$
Introduce MI, from R $\text{imputed}_matrix_mi = np.ones((10000, 100000))$
$\text{imputed}_matrix_mi = np.concatenate((data_array, data_array_2), axis = 1)$
$\text{observation}_means_mi = np.ones((10000,))observation_variances_mi = np.ones((10000,))observation_means_mi = np.nanmean(imputed_matrix_mi, axis = 1)observation_variances_mi = np.nanvar(imputed_matrix_mi, axis = 1)std_evmi = np.sqrt(observation_variances_mi)$
$\text{margin}_of_error = z * std_evmiMarginoferrorlower_bound = observation_means_mi - margin_of_errorupper_bound = observation_means_mi + margin_of_error$
For 1000 case $\text{imputation}_result_mi = pd.read_excel("C :$
$Users$
$wkoomen001$
$Documents$
$thesis_results.xlsx")imputed_matrix_mi = pd.read_excel("C :$
$Users$
$wkoomen001$
$Documents$
$thesis$
$imputed_matrix_mi.xlsx")imputed_matrix_mi = imputed_matrix_mi.to_numpy(dtype = np.float64)observation_means_mi[:,] = imputation_result_mi['Mean']observation_means_mi =$

$np.array(observation_means_mi)observation_variances_mi[:,] = imputation_result_mi['Variance']st_dev_mi = np.sqrt(observation_variances_mi)$

lower$_b$ound $= imputation_result_mi['CI_Lower']upper_bound = imputation_result_mi['CI_Upper']$

Create a DataFrame for the confidence intervals confidence$_i$ntervals$_n$ormal$_m$i $= pd.DataFrame('lower_bound' : lower_bound,' upper_bound' : upper_bound, index = df.index)$

calculate MAE for MI def calculate$_i$nsample$_m$se$(actual, predicted)$: $actual = np.array(actual)predicted = np.array(predicted)onlyfirst2000, outofsampleactual = actual[: 2000]predicted = predicted[: 2000]SAE = 0foriinrange(len(predicted)) : SAE+ = np.abs(actual[i] - predicted[i])MAE = SAE/len(predicted)$

mse = np.mean((actual-predicted) ** 2)

return MAE mse$_m$i $= calculate_insample_mse(observed_values, observation_means_mi)$

Make the covariances

def calculate$_c$ovariances$(matrix1, matrix2)$: $covariances = []foriinrange(matrix1.shape[0]) : Identifynon - NApositionsforthecurrentrowvalid_positions = np.isnan(matrix1[i,:]) np.isnan(matrix2[i,:])$

Extract the valid data points valid$_d$ata1 $= matrix1[i, valid_positions]valid_data2 = matrix2[i, valid_positions]$

Only compute covariance if there are valid points if len(valid$_d$ata1) $> 1andlen(valid_data2) > 1 : cov_matrix = np.cov(valid_data1, valid_data2, ddof = 1)covariances.append(cov_matrix[0, 1])else : covariances.append(np.nan)AppendNaNifnotenoughvalidpointsreturnnp.array(covariances)$

Example usage with your matrices Assuming imputed$_m$atrix$_r$idge, $imputed_matrix_knn, imputed_matrix_rfarede$

$calculate_covariances(imputed_matrix_ridge, imputed_matrix_knn)covariances_ridge_rf =$

$calculate_covariances(imputed_matrix_ridge, imputed_matrix_rf)covariances_knn_rf =$

$calculate_covariances(imputed_matrix_knn, imputed_matrix_rf)covariances_mi_ridge =$

$calculate_covariances(imputed_matrix_mi, imputed_matrix_ridge)covariances_mi_rf =$

$calculate_covariances(imputed_matrix_mi, imputed_matrix_rf)covariances_mi_knn =$

$calculate_covariances(imputed_matrix_mi, imputed_matrix_knn)$

weights using in sample mae maes = np.array([mse$_r$idge, $mse_knn, mse_rf, mse_mi])theta = 0.0001exp_neg_theta_weights = np.exp(-theta * maes)weights = exp_neg_theta_weights/np.sum(np.exp(-theta * maes))$

individuals = np.column$_s$tack$((observation_means_ridge, observation_means_knn, observation_means_rf, observatio$

$np.matmul(individuals, weights)var_y = w1 * var_1 + w2 * var_2 + w3 * var_3 + 2 * w1 * w2 * cov(1, 2) + 2 * w1 * w3 * cov(1, 3) + 2 * w2 * w3 * cov(2, 3)var_comb = weights[0] * weights[0] * observation_variances_ridge + weights[1] * weights[1] * observation_variances_knn + weights[2] * weights[2] * observation_variances_rf + weights[3] * weights[3] * observation_variances_mi + 2 * weights[0] * weights[1] * covariances_ridge_knn + 2 * weights[0] * weights[2] * covariances_ridge_rf + 2 * weights[0] * weights[3] * covariances_mi_ridge + 2 * weights[1] * weights[2] * covariances_knn_rf + 2 * weights[1] * weights[3] * covariances_mi_knn + 2 * weights[2] * weights[3] * covariances_mi_rfCalculatethestandarddeviationforeachobservationst_dev_comb = np.sqrt(var_comb)$

Calculate confidence intervals using normal distribution z = 2 Z-value for 95margin$_o$f$_e$rror $= z * st_dev_combMarginoferrorlower_bound = observation_means_combined -$

$margin_of_errorupper_bound = observation_means_combined + margin_of_error$
Create a DataFrame for the confidence intervals confidence$_intervals_normal_combined$ = $pd.DataFrame('lower_bound' : lower_bound,' upper_bound' : upper_bound, index = df.index)$
Evaluate
def violations(true$_value, lower_bound, upper_bound)$:
violations = [] for i in range(len(lower$_bound)) : violations.append(lower_bound[i] <= true_value[i] <= upper_bound[i])$
return violations
violations$_mi = violations(observed_values, confidence_intervals_normal_mi['lower_bound'], confidence_intervals_n$
violations$_mi = violations_mi[0 : 2000]$
nr$_violations_mi = len(violations_mi) - sum(violations_mi)$
bias$_mi = 0foriiinrange(len(violations_mi)) : bias_mi+ = observation_means_mi[i] - observed_values[i]$
bias$_mi = bias_mi/len(violations_mi)$
violations$_ridge = violations(observed_values, confidence_intervals_normal_ridge['lower_bound'], confidence_interv$
violations$_ridge = violations_ridge[0 : 2000]$
nr$_violations_ridge = len(violations_ridge) - sum(violations_ridge)$
bias$_ridge = 0foriiinrange(len(violations_ridge)) : bias_ridge+ = observation_means_ridge[i] - observed_values[i]$
bias$_ridge = bias_ridge/len(violations_ridge)$
violations$_knn = violations(observed_values, confidence_intervals_normal_knn['lower_bound'], confidence_interval$
violations$_knn = violations_knn[0 : 2000]$
nr$_violations_knn = len(violations_knn) - sum(violations_knn)$
bias$_knn = 0foriiinrange(len(violations_knn)) : bias_knn+ = observation_means_knn[i] - observed_values[i]$
bias$_knn = bias_knn/len(violations_knn)$
violations$_rf = violations(observed_values, confidence_intervals_normal_rf['lower_bound'], confidence_intervals_n$
violations$_rf = violations_rf[0 : 2000]$
nr$_violations_rf = len(violations_rf) - sum(violations_rf)$
bias$_rf = 0foriiinrange(len(violations_rf)) : bias_rf+ = observation_means_rf[i] - observed_values[i]$
bias$_rf = bias_rf/len(violations_rf)$
violations$_combined = violations(observed_values, confidence_intervals_normal_combined['lower_bound'], confider$
violations$_combined = violations_combined[0 : 2000]$
nr$_violations_combined = len(violations_combined) - sum(violations_combined)$
bias$_combined = 0foriiinrange(len(violations_combined)) : bias_combined+ = observation_means_combined[i] - observed_values[i]$
bias$_combined = bias_combined/len(violations_combined)$
def calculate$_outsample_mse(actual, predicted) : actual = np.array(actual)predicted = np.array(predicted)onlyfirst2000, insampleactual = actual[: 2000]predicted = predicted[: 2000]SAE = 0foriiinrange(len(predicted)) : SAE+ = np.abs(actual[i] - predicted[i])MAE = SAE/len(predicted)$

mse = np.mean((actual-predicted) ** 2)
return MAE
$mse_ridge = calculate_outsample_mse(observed_values, observation_means_ridge) mse_knn = calculate_outsample_mse(observed_values, observation_means_knn) mse_rf = calculate_outsample_mse(observed_values, observation_means_rf) mse_mi = calculate_outsample_mse(observed_values, observation_means_mi) mse_combined = calculate_outsample_mse(observed_values, observation_means_combined)$
$\text{def average}_confidence_interval_length(df) : """CalculatetheaveragelengthofconfidenceintervalsfromaDataF$
Parameters: df (pd.DataFrame): DataFrame containing the confidence intervals. $\text{lower}_col(str)$: $Nameofthecolumnwiththelowerbounds.\text{upper}_col(str)$: $Nameofthecolumnwiththeupperbounds.$
Returns: float: Average length of confidence intervals. """ Calculate the length of each confidence interval df = df[:2000] $\text{interval}_length = df['upper_bound'] - df['lower_bound']$
Compute the average length of the intervals $\text{average}_length = interval_length.mean()$
return average_length
Calculate the average confidence interval length $\text{average}_length_ridge = average_confidence_interval_length(confidence_intervals_normal_ridge) average_length_knn = average_confidence_interval_length(confidence_intervals_normal_knn) average_length_rf = average_confidence_interval_length(confidence_intervals_normal_rf) average_length_mi = average_confidence_interval_length(confidence_intervals_normal_mi) average_length_combined = average_confidence_interval_length(confidence_intervals_normal_combined)$
import matplotlib.pyplot as plt plt.hist($\text{imputed}_matrix_rf[10], bins =' auto', alpha = 0.7, color =' blue'$)
Plotting plt.figure(figsize=(20, 6)) plt.errorbar(df.index[1:101], $\text{observation}_means_combined[1 : 101], yerr = (confidence_intervals_normal_combined['upper_bound'] - observation_means_combined)[1 : 101], fmt =' o', markersize = 5, capsize = 5, color =' 7D7D7D')Plottheobservedvaluesinredplt.scatter(df.index[1 : 101], observed_values[1 : 101], color =' D04A02', marker =' x', label =' Observed', zorder = 5)$
plt.xlabel('Observation Index') plt.ylabel('Mean Value') plt.title('Mean Value with 95plt.grid(True) plt.show()
-*- coding: utf-8 -*- """ Created on Mon Jul 22 15:55:18 2024
@author: wkoomen001 """
import pandas as pd import numpy as np import time as time import matplotlib.pyplot as plt import seaborn as sns import os from $\text{sklearn.linear}_model import LogisticRegression from sklearn.model_selection import train_test_split from sklearn.m$
Load and clean data df = $\text{pd.read}_excel("C :$
$Users$
$koome$
$Documents$
$Vanoudepc$
$Master$
$Thesis$

$2010Q4_sub100.xlsx")df = df.drop(columns = df.columns[df.isna().sum() > 10000])df = df.dropna()$
for column in df.columns: print('column name is', column) print(df[column].unique())
drop = [] for column in df.columns: if len(df[column].unique()) ¡= 1: print(column) drop.append(column)
df = df.drop(columns=drop)
$drop_columns1 = ['ServicerName','SellerName','MaturityDate','PropertyState','MetropolitanStatistical$
$df.drop(drop_columns1, axis = 1)$
df['Current Loan Delinquency Status'].$value_counts()df.loc[df['CurrentLoanDelinquencyStatus'] >$
$1,'CurrentLoanDelinquencyStatus'] = 1df['CurrentLoanDelinquencyStatus'].value_counts()$
$df_model = df.drop(columns =' CurrentLoanDelinquencyStatus', axis = 1)y = df['CurrentLoanDelinquencyStatus'].astype(int)X = pd.get_dummies(df_model, dtype = int)$
$X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 42)$
Check multicollinearity using VIF and remove highly correlated features def $calculate_vif(X) : vif = pd.DataFrame()vif["features"] = X.columnsvif["VIF"] = [variance_inflation_factor(X.values, i)foriinrange(X.shape[1])]returnvif$
def $remove_high_vif_features(X, threshold = 9.5) : vif = calculate_vif(X)whilevif["VIF"].max() > threshold : feature_to_remove = vif.sort_values("VIF", ascending = False).iloc[0]["features"]print(f'Removingfeaturefeature_to_removewith$
$X.drop(columns = [feature_to_remove])vif = calculate_vif(X)returnX$
$X_train = remove_high_vif_features(X_train)$
Backward selection process for logistic regression def $backward_selection(X, y) : initial_features = X.columns.tolist()whilelen(initial_features) > 0 : X_1 = sm.add_constant(X[initial_features])model = sm.Logit(y, X_1).fit(disp = 0)p_values = model.pvalues.iloc[1 :]Ignorethep - valueoftheconstanttermmax_pvalue = p_values.max()ifmax_pvalue > 0.05 : excluded_feature = p_values.idxmax()initial_features.remove(excluded_feature)print(f'Removedfeature : excluded_featurewithp - value : max_pvalue')else : breakreturninitial_features$
$selected_features = backward_selection(X_train, y_train)$
Ensure $X_trainandX_testcontainonlytheselectedfeatures, includingthetargetvariableforimputationifnecessar$
$To - Income(DTI)'notinselected_features : selected_features.append('Debt - To - Income(DTI)')$
$X_train_selected = X_train[selected_features]X_test_selected = X_test[selected_features]$
Handle class imbalance using SMOTE smote = SMOTE(random$_state = 42)X_train_balanced, y_train_balanced = smote.fit_resample(X_train_selected, y_train)$
Fit the logistic regression model with the selected features model = LogisticRegression(class$_weight =' balanced')model.fit(X_train_balanced, y_train_balanced)$
Function for bootstrap imputation def $bootstrap_imputed_matrix_combined(X_trainset, X_testset, variable_name, m$
$imputed_matrix_rf = np.full((len(X_testset), m), np.nan)sample_size = int(0.1 * (len(X_trainset)))Samplesizeforbootstrap$
for imp in range(m): Bootstrap resampling $bootstrap_data = np.random.choice(X_trainset.index, size = sample_size, replace = False)X_train =$

$X_trainset.loc[bootstrap_data].copy()Y_train = X_train[variable_name]X_train = X_train.drop(columns = variable_name, axis = 1)X_test = X_testset.drop(columns = variable_name, axis = 1)$

Random Forest imputation $rf_model = RandomForestRegressor()rf_model.fit(X_train, Y_train)y_missing_predicte$

$rf_model.predict(X_test)imputed_matrix_rf[:, imp] = y_missing_predicted_rf$

return $imputed_matrix_rf$

Impute the 'Debt-To-Income (DTI)' column in $X_testusingthetrainingdataimputed_matrix_rf = bootstrap_imputed_matrix_combined(X_train_selected, X_test_selected,' Debt - To - Income(DTI)', m = 100)$

Compute the mean of the imputations for the test set $observation_means_rf = np.nanmean(imputed_matrix_rf, axis = 1)observation_variances_rf = np.nanvar(imputed_matrix_rf, axis = 1)std_ev_rf = np.sqrt(observation_variances_rf)Calculateconfidenceintervalsusingnormaldistributionz = 2Z - valuefor95margin_of_error = z * std_ev_rfMarginoferrorlower_bound = observation_means_rf - margin_of_errorupper_bound = observation_means_rf + margin_of_error$

Apply the imputed values to the test set $X_testimputed = X_test.copy()X_testimputed['Debt - To - Income(DTI)'] = observation_means_rfX_testlower_bound = X_test.copy()X_testlower_bound['Debt - To - Income(DTI)'] = upper_bound$

Predict using the trained logistic regression model $y_pred_proba_true = model.predict_proba(X_test[selected_features])[:, 1]$

Compute ROC curve and ROC AUC for the true values $fpr_true, tpr_true, thresholds_true = roc_curve(y_test, y_pred_proba_true)roc_auc_true = roc_auc_score(y_test, y_pred_proba_true)$

Plot ROC curve for the true values plt.figure(figsize=(12, 8)) $plt.plot(fpr_true, tpr_true, color =' blue', lw = 2, label = f'ROCCurve(True)(AUC = roc_auc_true : .2f)')plt.plot([0, 1], [0, 1], color =' gray', linestyle =' --')plt.xlabel('FalsePositiveRate')plt.ylabel('TruePositiveRate')plt.title('ReceiverOperatingCharacteristi lowerright')plt.grid(True)plt.show()$

Find optimal threshold using the ROC curve def $find_optimal_threshold(fpr, tpr, thresholds)$: $optimal_idx = np.argmax(tpr - fpr)returnthresholds[optimal_idx]$

$optimal_threshold_true = find_optimal_threshold(fpr_true, tpr_true, thresholds_true)print(f'Optimalthresholdfor optimal_threshold_true')$

Apply the optimal threshold to make final predictions

$y_pred_proba_imputed = model.predict_proba(X_test_imputed[selected_features])[: , 1]y_pred_proba_lower_bound = model.predict_proba(X_test_lower_bound[selected_features])[:, 1]$

$y_pred_true = (y_pred_proba_true >= optimal_threshold_true).astype(int)y_pred_imputed = (y_pred_proba_imputed >= optimal_threshold_true).astype(int)y_pred_lower_bound = (y_pred_proba_lower_bound >= optimal_threshold_true).astype(int)$

$mean_ypred_true = np.nanmean(y_pred_true, axis = 0)mean_ypred_imputed = np.nanmean(y_pred_imputed, axis = 0)mean_ypred_lower_bound = np.nanmean(y_pred_lower_bound, axis = 0)print("MeanpredictedPDfortruevalues : ")print(mean_ypred_true)print("MeanpredictedPDforimputedvalues : ")print(mean_ypred_imputed)print("MeanpredictedPDforlowerboundvalues :$

")$print(mean_{yp}red_{l}ower_{b}ound)$
Perform paired t-tests ttest$_{t}rue_{v}s_{i}mputed = ttest_{r}el(y_{p}red_{t}rue, y_{p}red_{i}mputed)ttest_{t}rue_{v}s_{l}ower_{b}ound = ttest_{r}el(y_{p}red_{t}rue, y_{p}red_{l}ower_{b}ound)$
print("-test results (True vs Imputed):") print(f"t-statistic: ttest$_{t}rue_{v}s_{i}mputed.statistic, p - value : ttest_{t}rue_{v}s_{i}mputed.pvalue$")
print("-test results (True vs Lower Bound):") print(f"t-statistic: ttest$_{t}rue_{v}s_{l}ower_{b}ound.statistic, p - value : ttest_{t}rue_{v}s_{l}ower_{b}ound.pvalue$")
Print classification reports for each prediction print("Classification Report for True Values:")
print(classification$_{r}eport(y_{t}est, y_{p}red_{t}rue)$)
print("Classification Report for Imputed Values:") print(classification$_{r}eport(y_{t}est, y_{p}red_{i}mputed)$)
print("Classification Report for Lower Bound of 5print(classification$_{r}eport(y_{t}est, y_{p}red_{l}ower_{b}ound)$)