Handling Missing Data: Imputation Methods for Null Values

Yael Dahari (325166510), Arbel Tepper (209222272)

March 2025

Abstract

The handling of missing data is a crucial challenge in data analysis, as incorrect imputation can lead to biased conclusions and reduced model performance. Our solution is a tool that allows users to select an attribute for imputation, apply multiple imputation methods, and evaluate their effectiveness using various metrics. It also provides explanations for each method, helping users make informed decisions without the need for manual implementation. Experimental results show that our approach consistently outperforms the baseline random imputation method, ensuring more accurate and reliable data restoration, as well as improved model performance.

1 Introduction

1.1 Problem Description

Our goal is to improve the process of imputing missing values in datasets by providing users with multiple imputation options, along with an evaluation of their impact on model performance. This will enable users to make informed decisions about which method to use.

While working on previous assignments, we encountered the challenge of handling missing values, a process that required significant time and effort. We wanted an automated tool that could test different imputation methods and summarize their effects, allowing us to choose the most effective approach without trial and error. This project aims to create exactly that solution.

In class, we discussed various methods for imputing missing values and their potential impact on model performance. This project builds on those concepts by exploring how to quantitatively and qualitatively assess different imputation techniques.

Many users may not fully understand how different methods influence accuracy, fairness, and explainability. As a result, an imputation choice that seems beneficial may actually degrade model performance. Our project seeks to bridge this gap by automating the comparison of imputation methods, saving users time while ensuring that they select the most suitable approach for their data.

1.2 Solution Overview

Our solution automates and improves the process of imputing missing values in datasets by systematically testing multiple imputation methods and evaluating their impact on model performance.

1. **Dataset Input & Missing Data Analysis**
   (a) The user uploads a dataset and our tool automatically analyzes the missing values.
   (b) A summary is presented that details the attributes with missing values, the number of affected samples, and their percentage relative to the size of the dataset.
2. **User Selection of Attributes for Imputation**
   (a) The user selects the attributes they wish to impute.
3. **Baseline Imputation for Benchmarking**
   (a) As a naive baseline, missing values are filled with random values within the attribute's observed range (min-max).
   (b) This serves as a reference point to assess whether more sophisticated imputation methods improve model performance.

4. **Application of Multiple Imputation Techniques**
   (a) We apply several imputation strategies, including:
      i. **Mean & Median Imputation** – Filling missing values with the attribute's mean or median.
      ii. **K-Nearest Neighbors (KNN) Imputation** – Estimating missing values based on the nearest neighbors.
      iii. **Regression Imputation (LR)** – Predicting missing values using regression models trained on non-missing data.
      iv. **Row Deletion** – Removing samples with missing values as a last-resort option.
5. **Performance Evaluation & Comparison**
   (a) We evaluate each imputation method by training a model on the imputed data and computing multiple performance metrics, including:
      i. **R^2 Score** – Measures how well the model explains variance.
      ii. **Mean Absolute Error (MAE)** – Average absolute difference between predicted and actual values.
      iii. **Mean Absolute Percentage Error (MAPE)** – Measures error as a percentage.
      iv. **Mean Squared Error (MSE) & Root Mean Squared Error (RMSE)** – Penalize larger errors more heavily.
   (b) The results provide a quantitative basis for comparing the imputation methods.
6. **User Guidance & Final Imputation**
   (a) The tool presents the evaluation results alongside explanations of the strengths and weaknesses of each method.
   (b) The user selects their preferred imputation method(s), which are then applied to the entire dataset.
   (c) The final imputed dataset is saved to a location specified by the user.

Key Benefits

- **Automation** – Reduces the manual effort required for testing imputation techniques.
- **Data-Driven Decision Making** – Enables users to make informed choices based on empirical results.
- **Flexibility** – Supports multiple imputation methods and user-defined preferences.
- **Improved Model Performance** – Ensures that missing data handling contributes positively to downstream tasks.

This structured approach provides users with a practical and efficient tool for handling missing data, ultimately improving the data science workflow.

2 Experimental Evaluation

To assess the effectiveness of our imputation methods, we conducted experiments on four different datasets using six missing data imputation algorithms. Since the original datasets did not contain missing values, we artificially introduced missing data using the *Missing Completely at Random (MCAR)* model [3]. This approach allows us to objectively evaluate imputation quality by comparing the imputed values with the original ones.

For each dataset, we randomly selected attributes to introduce missing values and varied the percentage of missing data. The artificially generated missing values enable us to quantify imputation accuracy, ensuring a fair and controlled evaluation of different methods.

2.1 Additional Evaluation Metric: Similarity Score

In addition to standard evaluation metrics (such as **R^2, MAE, MSE, RMSE, and MAPE**, discussed previously), our approach enables a **unique similarity score**, which quantifies how accurately missing values were imputed compared to the original data.

Since we artificially introduced missing values, we know the true values and can directly measure how close the imputations are. This **similarity score** is scaled between **0 and 1**, where:

- **1.0** indicates perfect recovery (imputed values exactly match the original ones).
- **0.0** indicates a total mismatch.

For **categorical attributes**, the similarity score is simply the fraction of correctly imputed values (i.e., those matching the original values).

For **numerical attributes**, we normalize the **Mean Absolute Error (MAE)** by the attribute's range:

$$\text{Normalized MAE} = \frac{\text{MAE}}{\max - \min}$$

Since lower MAE indicates better performance, we define the final similarity score as:

$$SimilarityScore = 1 - NormalizedMAE$$

This ensures that the score remains within the **[0,1]** range, making it comparable across different attributes, which may have values of different scales.

2.2 Handling Special Cases

It is important to note that the **"drop rows"** method does not perform imputation but rather removes samples with missing values. As a result, it is **not possible** to compute a similarity score for this method, and instead we'll write NA.

2.3 Reproducibility Considerations

Since missing values are introduced randomly, results may vary between runs. While the trends observed in our experiments provide meaningful insights, any conclusions drawn should be interpreted with caution and verified through additional testing.

2.4 First Dataset: Movies Revenue

The Movies Revenue dataset contains 34 features related to movies, including budget, title, runtime, crew, cast, and more, with the goal of predicting movie revenue.

- **Dataset dimensions**: (5368, 34)
- **Target**: Revenue
- **Attributes with missing values introduced**: budget, original_language, vote_count, popularity
- **Percentage of missing values per attribute**: 27%, 32%, 27%, and 25%, respectively

After introducing missing values, we applied six different imputation methods and evaluated their impact on prediction performance. Below are the scores obtained for the **budget** attribute:

	R^2	MSE	RMSE	MAPE	MAE	Similarity
Random	0.448	2.267702e+16	1.505889e+08	5019295.022%	9.855290e+07	0.887
Mean	0.709	1.195051e+16	1.093184e+08	3720342.997%	5.736363e+07	0.977
Median	0.705	1.212208e+16	1.101003e+08	2721735.336%	5.731354e+07	0.979
Frequent	0.706	1.207491e+16	1.098859e+08	2930868.433%	5.720527e+07	0.979
KNN	0.709	1.195051e+16	1.093184e+08	3720342.997%	5.736363e+07	0.977
LR	0.694	1.256332e+16	1.120862e+08	2374022.588%	5.933177e+07	1.000
Drop	0.737	1.114388e+16	1.055646e+08	8554331.544%	5.408191e+07	NA

2.4.1 Analysis of Results

- **Random imputation performed the worst**, yielding the lowest **R^2 score (0.448)**, indicating that filling missing values randomly severely degrades model performance.
- **All other imputation methods significantly improved R^2**, with values clustering around **0.7**, suggesting they provide more reliable estimations.
- **Error scores (MSE, RMSE, MAPE, MAE) are extremely high** across all methods. This is expected since **revenue is measured in millions of dollars**, meaning even small absolute differences translate into large error values. If the dataset were **normalized**, these errors would likely be much smaller.
- The **similarity score**, which represents how well the imputed values match the original data, is relatively high even for random imputation (0.887). This suggests that the budget values in the dataset are likely distributed normally, making random guesses sometimes reasonably close to actual values.
- **Mean and KNN imputation performed the best overall**, yielding high similarity scores ($\tilde{0}$.977) while maintaining solid predictive performance (R^2 $\tilde{0}$.709). However, given the principle of preferring simpler models over complex ones, Mean imputation is the preferable choice since it is computationally cheaper while performing just as well as KNN.

2.5 Second Dataset: Laptop Price

The Laptop Price dataset contains 11 attributes related to laptops, such as Memory, RAM, CPU, and more, with the goal of predicting laptop price.

- **Dataset dimensions**: (1303, 11)
- **Target**: Price
- **Attributes with missing values introduced**: Cpu, Company_Cpu, ScreenResolution
- **Percentage of missing values per attribute**: 26%, 17%, and 20%, respectively

We evaluated different imputation methods on the **Company_Cpu** attribute and obtained the following results:

	R^2	MSE	RMSE	MAPE	MAE	Similarity
Random	0.345	400863.157	633.138	60.607%	492.426	0.911
Mean	0.409	362041.728	601.699	53.669%	456.028	0.972
Median	0.412	359996.010	599.997	53.63%	454.635	0.974
Frequent	0.417	356953.755	597.456	53.97%	452.950	0.959
KNN	0.409	362041.728	601.699	53.669%	456.028	0.972
LR	0.417	356952.813	597.455	53.971%	452.951	1.000
Drop	0.395	382960.707	618.838	52.246%	462.653	NA

2.5.1 Analysis of Results

- **Minimal improvement in R^2:** Unlike the previous dataset, most imputation methods yield only a small R^2 improvement ($\tilde{0}$.05) over random imputation, suggesting that missing values in Company_Cpu do not significantly impact model performance.
- **Random imputation performs reasonably well**, with a similarity score of 0.911, indicating that the original data distribution is not drastically altered by missing values.
- **Frequent imputation emerges as the best choice**, as it provides the highest R^2 (0.417), while being simple and computationally efficient. Given the marginal performance difference between more complex methods like KNN or Linear Regression, opting for a fast, non-parametric approach like frequent imputation is preferable in this case.

2.6 Third Dataset: Cars Price

The Car Price dataset contains 32 features related to cars, such as age, number of cylinders, brand, accident history, and title status, with the goal of predicting car price.

- **Dataset dimensions**: (188,533, 32)
- **Target**: Price
- **Attributes with missing values introduced**: mileage, horse_power, brand, age
- **Percentage of missing values per attribute**: 28%, 26%, 32%, and 30%, respectively

We evaluated different imputation methods on the **brand** attribute and obtained the following results:

	R^2	MSE	RMSE	MAPE	MAE	Similarity
Random	0.475	0.605	0.778	308.944%	0.571	0.96
Mean	0.475	0.605	0.778	307.872%	0.570	0.97
Median	0.475	0.605	0.778	307.875%	0.570	0.97
Frequent	0.475	0.605	0.778	307.895%	0.570	0.97
KNN	0.475	0.605	0.778	307.872%	0.570	0.97
LR	0.472	0.608	0.780	309.581%	0.572	1.000
Drop	0.484	0.590	0.768	299.941%	0.565	NA

2.6.1 Analysis of Results

- **All imputation methods yield nearly identical results**, indicating that missing values in brand do not significantly impact model performance.
- **Potential reasons for identical results**:
  - The dataset's structure may be such that mean, median, and frequent values are very similar, leading to indistinguishable imputation effects.
  - The dataset is too complex for the Linear Regression model used for imputation, as seen in the high error metrics (MSE = 0.6, MAE = 0.5).
- Linear Regression achieves perfect restoration (similarity score = 1.000), yet the error metrics remain high. This suggests that the problem lies in the intrinsic complexity of the data rather than the imputation method itself—the relationships in the data do not follow a simple linear pattern.
- **An unexpected insight**: Since all imputation methods perform similarly, we gain a bonus understanding of the dataset's structure—implying that brand does not introduce significant variation in car price prediction.

2.7 Fourth Dataset: Avocado Average Price

The Avocado Average Price dataset contains 12 features related to avocados, such as type, year, and region, with the goal of predicting average avocado price.

- **Dataset dimensions**: (18,249, 12)
- **Target**: AveragePrice
- **Attributes with missing values introduced**: TotalBags, type, year
- **Percentage of missing values per attribute**: 28%, 31%, and 27%, respectively

We evaluated different imputation methods on the **type** attribute and obtained the following results:

	R^2	MSE	RMSE	MAPE	MAE	Similarity
Random	0.244	0.121	0.348	20.896%	0.272	0.840
Mean	0.294	0.113	0.336	20.103%	0.262	0.843
Median	0.229	0.123	0.351	21.184%	0.273	0.846
Frequent	0.229	0.123	0.351	21.184%	0.273	0.846
KNN	0.294	0.113	0.336	20.103%	0.262	0.843
LR	0.124	0.140	0.374	23.142%	0.297	1.000
Drop	0.403	0.096	0.310	17.995%	0.238	NA

2.7.1 Analysis of Results

- **Median and Frequent Value Imputation:** These methods yield identical results, implying that the most frequent value in the dataset is likely the median.
- **Mean and KNN Imputation:** These methods also produce identical scores, indicating that KNN is effectively approximating the mean in this case.
- **Linear Regression Imputation:**
  - Despite achieving a perfect restoration (Similarity = 1.000), it has the lowest R^2 (0.124) and highest error metrics (MSE = 0.140, MAE = 0.297).
  - This suggests that the dataset does not exhibit a strong linear relationship, making LR unsuitable for imputation.
- **Dropping Rows with Missing Values:**
  - This method significantly outperforms all others with an R^2 of 0.403, which is at least 0.1 higher than any other method.
  - This suggests that the presence of missing values introduces noise, and dropping them leads to a cleaner dataset and better predictive performance.

2.8 Evaluating Our Solution Against the Baseline

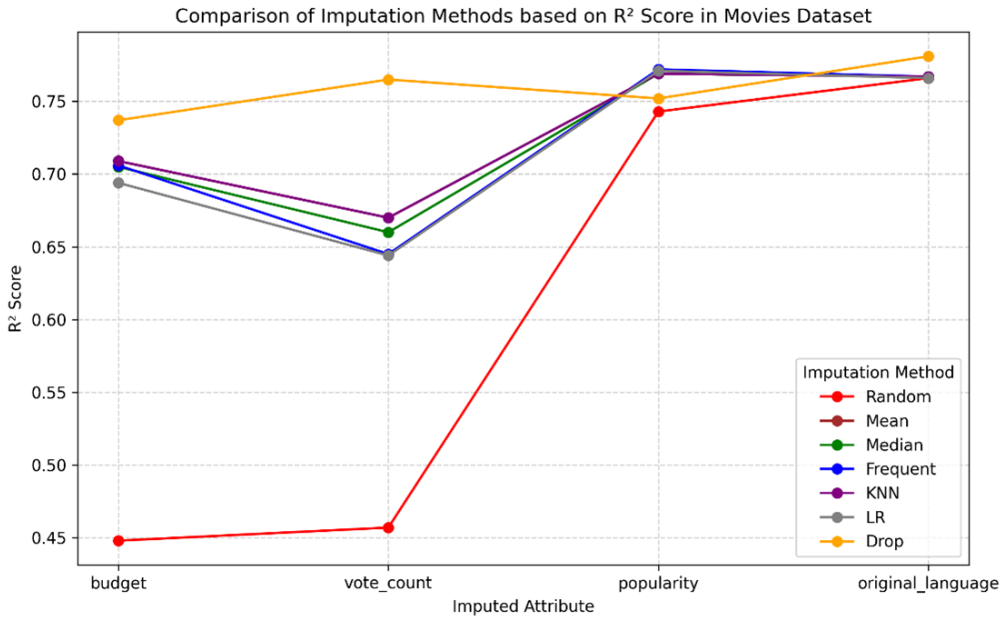
2.8.1 Comparing R^2 Scores Across Attributes

To demonstrate that our imputation tool outperforms the baseline method, we compared evaluation metrics across multiple attributes and datasets. However, since the target attribute was not normalized in most datasets, we opted not to use error-based evaluation metrics (such as MAE and MSE), as they would not provide a clear comparison. Instead, we focused on the **R^2 score** and the **similarity score**:

- The **R^2 score** measures how well the model predicts the target variable using the imputed dataset.
- The **similarity score** evaluates how closely the imputed values match the original missing values.

By analyzing these two metrics, we assess both the quality of the imputation itself and its impact on model performance.

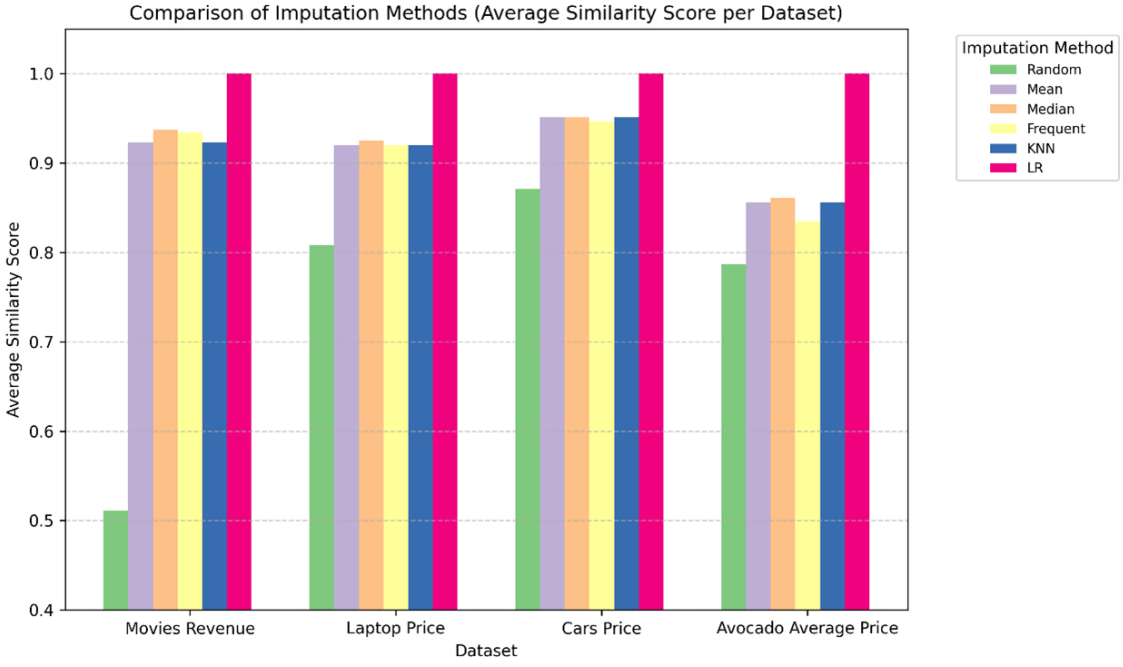
To visualize this, we plotted the **R^2 scores for each imputed attribute in the Movies dataset**, comparing the results across different imputation methods. Each method is represented by a different color, and the baseline **random imputation** is included for comparison.



As the plot shows, **the random imputation method consistently produces the lowest R^2 scores across all attributes**, confirming that our tool offers superior imputation strategies. While the improvement over the baseline is modest for some attributes (e.g., *popularity* and *original_language*), for others (*budget* and *vote_count*), the improvement is significant, highlighting the effectiveness of our approach.

2.8.2 Comparing Average Similarity Scores Across Datasets

To further analyze the performance of our imputation methods, we calculated the **average similarity score per dataset** for each method. This means that for each dataset, we averaged the similarity scores across all attributes and displayed the results in a bar chart, where each method is represented by a different color. The **Drop method** was excluded, as previously mentioned, since it does not allow for a similarity score calculation.



From the chart, we observe several key trends:

1. **Similar performance among Mean, Median, Frequent, and KNN** These methods produce nearly identical average similarity scores, suggesting that while KNN is computationally more complex, it does not necessarily yield better data restoration than simpler methods.

2. **LR consistently outperforms all methods** The **Linear Regression method achieves the highest similarity scores across all datasets**, indicating that the missing attributes may exhibit **co-linearity** with the rest of the data. This suggests that leveraging relationships between attributes significantly improves imputation quality.

3. **Random imputation consistently performs the worst** The baseline **random method has the lowest similarity scores across all datasets**. While some of the previous tables showed relatively high similarity scores for random imputation in individual attributes, this **dataset-wide average analysis provides a clearer picture**. It reinforces that **our imputation solutions consistently outperform the baseline method** over multiple attributes.

Overall, our results clearly demonstrate that our imputation methods significantly outperform the baseline random method. Based on the graphs shown, these findings validate that our imputation strategies significantly enhance data restoration quality and model performance compared to the baseline approach.

3 Related Work

Several studies have explored different approaches to handling missing values in datasets. Our solution builds on existing methods while introducing key enhancements to improve usability, flexibility, and practical applicability for data scientists.

One relevant work is *"Experimental Analysis of Methods for Imputation of Missing Values in Databases"* [1] . This paper evaluates multiple imputation algorithms, including Hot-Deck Imputation, Naïve Bayes, and Mean Imputation. Hot-Deck Imputation fills all missing values in a sample simultaneously, rather than handling each attribute separately. Since our approach focuses on per-attribute imputation, this method was not directly applicable to our problem. However, their methodology for evaluating imputation performance was insightful.

Another key reference is *"A Survey on Missing Data in Machine Learning"* [2], which provides a broad overview of imputation techniques ranging from simple methods (mean, median) to more complex approaches like Expectation-Maximization (EM), Support Vector Machines (SVM), and Decision Trees. While these advanced methods can be powerful, they introduce significant computational complexity. Given that our solution is designed for practical, real-time decision-making in the data science workflow, we opted for a more interpretable and computationally efficient set of imputation techniques. For example, since we are using Linear Regression as a learning model, we deemed it excessive to use an imputation method that is more complex than the model itself.

3.1 Key Differences & Contributions

While prior research has focused on proposing and evaluating different imputation methods, our solution provides a **personalized and interactive** approach:

1. **User-Centric Decision-Making** – Unlike most existing tools that apply a single imputation strategy to the entire dataset, our solution allows users to choose which attributes to impute and provides real-time comparisons of different methods.

2. **Evaluation & Explanation** – We not only compute performance metrics for each imputation method but also provide explanations about their advantages and drawbacks, helping users make informed decisions.

3. **Automated Dataset Processing** – Our tool produces a finalized, imputed dataset that users can directly use for further analysis, eliminating the need for manual data preprocessing.

3.2 Inspiration from Existing Work

One particularly valuable insight we gained from *"Experimental Analysis of Methods for Imputation of Missing Values in Databases"* [1] was their experimental setup. Instead of searching for datasets with naturally occurring missing values, they artificially introduced missing values using the *Missing Completely at Random (MCAR)* model. This ensures a controlled evaluation process, allowing us to compare the imputed values with the original ones. Inspired by this approach, we adopted a similar methodology to assess the effectiveness of our imputation techniques.

By building upon existing research and integrating a user-friendly evaluation process, our solution aims to bridge the gap between theoretical advancements in missing data imputation and practical implementation in data science workflows.

4 Future Work

Although our tool proved to be pretty useful, there is still room for improvement. Currently our experiment worked only with a Linear Regression model when training the imputed data and trying to predict the test data, but the next step would be to test it with different models and see how it behaves. When we'll introduce more complex learning models, we'll also be able to use more complex imputation models, which we refrained to use so far. We believe this would elevate our tool even more.

5 Conclusion

Through our analysis, we explored various imputation methods and evaluated their effectiveness in restoring missing data. Our findings indicate that sometimes complex and simple models perform similarly, meaning that the added complexity doesn't necessarily lead to better results. Additionally, our results show that Linear Regression consistently outperforms all other methods in restoring the missing values.

Most importantly, our comparison highlights that our solution consistently surpasses the baseline random imputation method. While random imputation appeared to achieve reasonable similarity scores for individual attributes, the average similarity scores across entire datasets reveal its limitations. This reinforces the importance of choosing a structured imputation strategy rather than relying on arbitrary value assignment.

Beyond these technical insights, this project emphasized the significance of providing users with both automated tools and interpretability. By integrating multiple methods and clear evaluation metrics, we enable data practitioners to make informed choices, balancing accuracy and efficiency in handling missing data.

6 Code and Data

The code for the automation and the graphs shown, as well as the datasets used are all available in our git repository under the final project directory. Link.

References

[1] A. Farhangfar, L. Kurgan, W. Pedrycz, "Experimental analysis of methods for imputation of missing values in databases", *Proceedings Volume 5421, Intelligent Computing: Theory and Applications II*, 2004.

[2] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, O. Tabona, "A survey on missing data in machine learning", *Journal of Big Data*, vol. 8, article 140, 2021.

[3] M. Wyss, "Understanding and Handling Missing Data", *Inwt blog*, 2020.