Vietnamese Germany University

Lab 1

Missing Value Handling Methods

Le Thanh Hai - 10421016

Introduction To Data Science And AI

Supervised by
Prof. Dr. Tung Kieu

April 14, 2023

Abstract

Missing values in real-world datasets are a common and challenging issue in data analysis and machine learning. Missing data can occur for various reasons, such as incomplete data collection, measurement errors, data entry mistakes, or intentional data withholding. Handling missing values is crucial as they can impact the accuracy and reliability of data analysis and machine learning models. This abstract reviews the prevalence and consequences of missing values in real-world datasets, discusses common techniques for handling missing data, including deletion, imputation, and advanced methods such as multiple imputation and machine learning-based approaches. Additionally, challenges and considerations in dealing with missing values, such as missing data patterns, missingness mechanisms, and potential biases, are addressed. Finally, the importance of careful handling of missing values in real-world datasets to ensure accurate and robust data analysis and model building is emphasized, and future research directions in handling missing data are outlined.

Table of Contents

Chapter 1

Introduction

1.1 Requirements

- Integrated Development Environment (IDE): PyCharm.
- Version Control: Git.
- Compiler Interpreter: Python 3 (Miniconda)
- Additional Libraries: Pandas, NumPy, SciPy, Matplotlib, Sklearn, Tensorflow.
- Data Sets: Iris

1.2 Ignoring Missing Value

Ignoring missing values, also known as complete case analysis or listwise deletion, is a method of handling missing data in which cases or records with any missing values are simply excluded from the analysis. In other words, any case that has at least one missing value is removed from the dataset, and only cases with complete data are included in the analysis.

Ignoring missing values is a relatively simple approach to handling missing data as it does not require any imputation or estimation techniques. It involves only using the cases with complete data, which eliminates the need to impute or fill in missing values. This method is commonly used when the percentage of missing values is low and is often used as a default method in many statistical software packages.

The main advantage of ignoring missing values is that it is straightforward to implement and does not require any assumptions about the data or imputation techniques. It can result in a complete dataset for analysis, and the analysis can be conducted on the available data without any imputed values. It can be particularly useful when the missing values are believed to be missing at random and not related to any systematic bias.

However, ignoring missing values also has some limitations. One major limitation is that it can lead to a loss of information if a large number of cases have missing values, resulting in reduced sample size and potentially biased results. This can reduce the statistical power of the analysis and may not accurately reflect the true population or sample characteristics. Additionally, ignoring missing values may introduce selection bias if the missingness is related to the variables of interest, leading to biased estimates and incorrect inferences.

1.3 Filling Mean Value

$$x_i = \frac{1}{N} \sum_{1}^{N} x \tag{1.1}$$

Mean imputation is a method of handling missing values where missing values are replaced with the mean (or average) of the available data for that variable. It is a simple imputation technique that involves calculating the mean of the observed values for a particular variable and replacing the missing values with this calculated mean. Equation (1.1)

The process of mean imputation typically involves the following steps:

1. Identify the variables with missing values: Determine which variables in the dataset have missing values that need to be imputed.
2. Calculate the mean: For each variable with missing values, calculate the mean of the observed (non-missing) values for that variable.
3. Replace missing values: Replace the missing values for each variable with the calculated mean.

Mean imputation is a straightforward method to handle missing values and has some advantages. One of the main advantages is its simplicity and ease of implementation, as it only requires basic calculations and does not involve complex modeling or estimation techniques. It can also help to retain the original sample size since no cases are excluded from the analysis, and a complete dataset is maintained after imputation.

However, mean imputation also has limitations. One major limitation is that it can introduce bias in the data, as it assumes that the missing values are missing at random and that the mean is a representative value for the missing values. This can result in biased estimates, especially if the missingness is related to any systematic pattern or variable of interest. Mean imputation also does not take into account the variability or distribution of the data, as it replaces missing values with a single value (the mean), which can result in an underestimation of the standard errors and inflated statistical significance.

1.4 Filling With Global Constant

Mean imputation is a method of handling missing values where missing values are replaced with the mean (or average) of the available data for that variable. It is a simple imputation technique that involves calculating the mean of the observed values for a particular variable and replacing the

missing values with this calculated mean.

The process of mean imputation typically involves the following steps:

1. Identify the variables with missing values: Determine which variables in the dataset have missing values that need to be imputed.
2. Choose a constant value: Select a fixed value, such as zero or any other predetermined constant, to be used as the imputed value for all missing values in the dataset.
3. Replace missing values: Replace the missing values for each variable with the chosen constant value.

Filling with a global constant is a simple and straightforward method to handle missing values, similar to mean imputation, and has some similar advantages. One of the main advantages is its simplicity and ease of implementation, as it does not require complex calculations or modeling techniques. It can also help to retain the original sample size, as no cases are excluded from the analysis, and a complete dataset is maintained after imputation.

However, filling with a global constant also has similar limitations to mean imputation. It can introduce bias in the data, as it assumes that the missing values are missing at random and that the chosen constant is a representative value for the missing values. This can result in biased estimates, especially if the missingness is related to any systematic pattern or variable of interest. Filling with a global constant also does not take into account the variability or distribution of the data, as it replaces missing values with a single value, which can result in an underestimation of the standard errors and inflated statistical significance.

1.5 Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on the Bayesian probability theorem, which calculates the probability of an event occurring given prior knowledge about related events.

Given an input sample $X = X_1, X_2, ..., X_n$ where x_i represent the i^{th} feature or atribute, and a set of classes $C = C_1, C_2, .., C_k$. The Naive Bayes classifier calculates the conditional probability of each class c in C given the input sample X, and selects the class with the highest probability as the predicted class for X. The formula for the conditional probability of a class given the input sample is given by eq. (1.2):

$$P(C|X) = \frac{P(C)P(X|C)}{P(X)} \tag{1.2}$$

where:

- $P(C|X)$ is the conditional probability of class c given the input sample X.
- $P(c)$ is the prior probability of class c (i.e., the probability of a sample belonging to class c without considering any features).

- $P(X|C)$ is the likelihood of the input sample X given class C (i.e., the probability of observing the features of X given that X belongs to class c)
- $P(X)$ is the probability of observing the input sample X (i.e., the normalization factor).

The "naive" assumption in Naive Bayes is that the features or attributes in X are conditionally independent given the class label, which simplifies the calculation of the likelihood P(X — c) as the product of the individual feature probabilities eq. (1.3):

$$P(X|c) = P(x_1|c) * P(x_2|c) * ... * P(x_n|c) \tag{1.3}$$

This assumption makes the Naive Bayes algorithm computationally efficient and allows it to work well with high-dimensional data. However, it may not always hold true in practice, as it assumes that the features are completely independent, which may not be the case in some real-world scenarios.

1.6 Min Max Normalization

MinMax normalization, also known as min-max scaling, is a data preprocessing technique used to rescale numerical features in a dataset to a specific range, typically between 0 and 1. It involves linearly transforming the original values of a feature to a new scale within the specified range.

The formula for min-max normalization is:

$$X_scale = \frac{x - x_{min}}{x_{min} - x_{max}} \tag{1.4}$$

1.7 Standard Normalization

Z normalization, also known as standardization or zero-mean normalization, is a data preprocessing technique used to transform numerical features in a dataset to have zero mean and unit variance. It involves subtracting the mean of the feature from each data point and then dividing by the standard deviation of the feature.

The formula for z normalization is:

$$X_scale = \frac{X - \mu}{\sigma} \tag{1.5}$$

Z normalization is commonly used in machine learning and data analysis to standardize features with different means and variances to a common scale. Standardization can be particularly useful for algorithms that are sensitive to the scale of input features, as it can prevent features with larger variances from dominating features with smaller variances during model training. This can be beneficial for various algorithms, including linear regression, logistic regression, support vector machines, and k-nearest neighbors, among others.

Chapter 2

Filling Missing Value Network

2.1 Network

Instead of filling those values manually, using network to learn and recognize the pattern which covers the meaning and relationship among data points in each feature. fig. 2.1

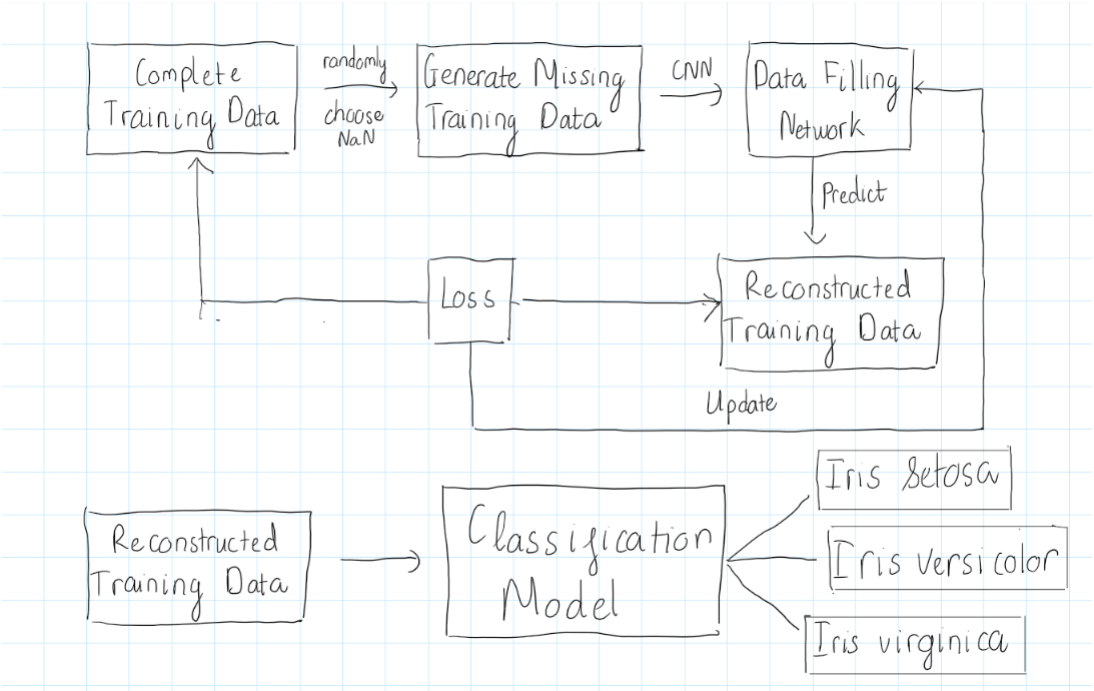


Figure 2.1: Data Filling Network

2.2 End-To-End Network

Before, I train the filling network and classification model independently which cost a lot of computational time and implementation. Hence, I combine both model into one which call end-to-end network [1] fig. 2.2

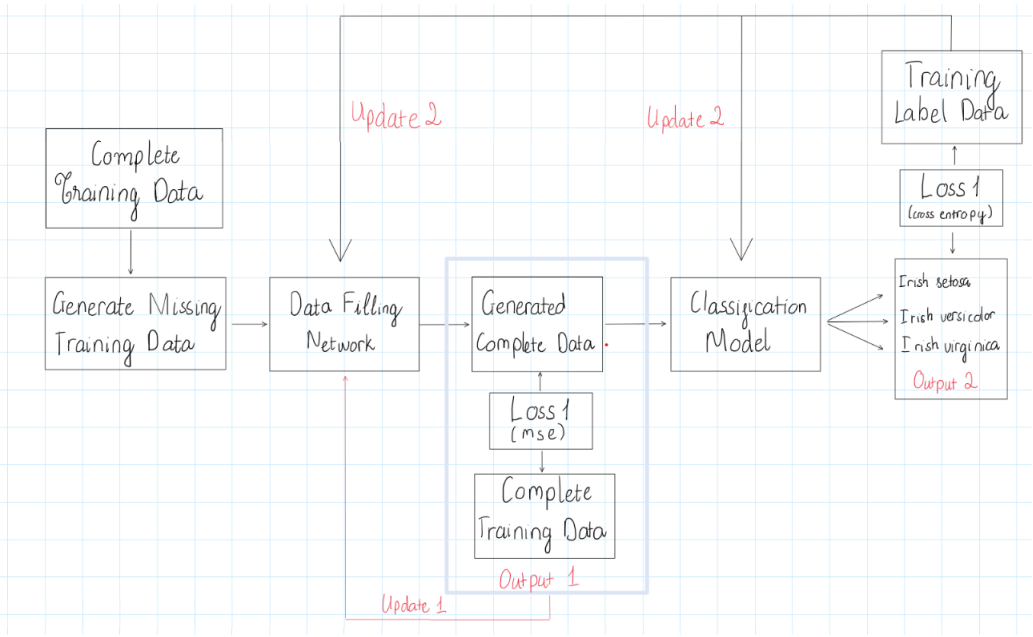


Figure 2.2: End-to-End Data Filling Network

2.3 Data Filling Network

A data filling network, also known as a data imputation or data completion network, is a type of machine learning model that is designed to fill in missing or incomplete data points in a dataset. Data filling networks are used in various domains where data is incomplete, such as finance, healthcare, marketing, and more.

Data filling networks typically leverage various machine learning techniques, such as deep learning, to learn patterns in the available data and use those patterns to predict and fill in missing values. These networks can handle different types of missing data, including missing values in numerical, categorical, or time-series data.

Data filling networks are trained on labeled datasets, where the model learns from examples of complete data to impute missing values in new, unseen data. Once trained, the model can be used to fill in missing values in real-world datasets, making it a valuable tool for data preprocessing and analysis tasks. [2]

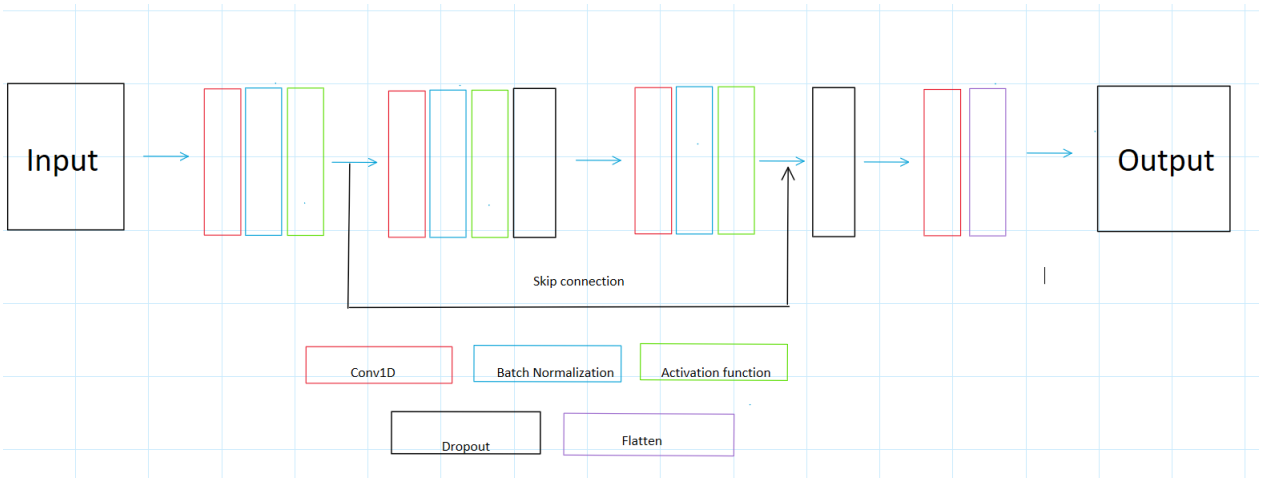


Figure 2.3: Data Filling Network

2.4 Loss Function

MSE (Mean Squared Error) is a commonly used loss function in data filling networks for regression tasks, including data imputation or data completion. It measures the average squared difference between the predicted values and the actual values. The formula for MSE is eq. (2.1):

$$MSE = \frac{1}{N}(y_{actual} - y_{pred})^2 \tag{2.1}$$

The MSE loss function is used in the training phase of data filling networks to quantify the discrepancy between the predicted values and the actual values. During training, the data filling network adjusts its parameters to minimize the MSE loss, which helps the model learn to impute missing values more accurately. Lower MSE values indicate better performance, as it means the predicted values are closer to the actual values.

2.5 Skip connection

Skip connections, also known as residual connections, are a type of connection used in neural networks that allow for the direct flow of information from one layer to a later layer, bypassing one or more intermediate layers. Skip connections were first introduced in the ResNet (Residual Network) architecture, which is a type of deep neural network.

In a typical neural network, information flows sequentially through the layers from input to output. However, in a neural network with skip connections, some of the input from a previous layer is added directly to the output of a later layer, forming a shortcut connection. This allows the network to "skip" over intermediate layers and pass information directly to subsequent layers. The output of the later layer is then obtained as the sum of the original output and the input from the previous layer.

Chapter 3

Experiment

3.1 Accuracy

3.1.1 No Normalization

Filling Method	Accuracy			
	5% nan	10% nan	15% nan	20% nan
Ignoring	93.33%	93.33%	96.67%	93.33%
Mean	93.33%	93.33%	93.33%	93.33%
Constant	96.67%	93.33%	96.67%	90%
Network	96.67%	96.67%	96.67%	93.33%
End-to-end	96.67 %	96.67%	96.67%	96.67%

Table 3.1: Accuracy of model without normalization

3.1.2 Min Max Normalization

Filling Method	Accuracy			
	5% nan	10% nan	15% nan	20% nan
Ignoring	90%	93.33%	86.67%	90%
Mean	80%	83.33%	80%	80%
Constant	86.67%	93.33%	76.67%	90%
Network	90%	90%	93.33%	90%
End-to-end	90%	93.33%	90%	90%

Table 3.2: Accuracy with Min Max Normalization

3.1.3 Standard Normalization

Filling Method	Accuracy			
	5% nan	10% nan	15% nan	20% nan
Ignoring	93.33%	90%	93.33%	90%
Mean	90%	90%	90%	90%
Constant	86.67%	90%	90%	90%
Network	96.67%	96.67%	96.67%	96.67%
End-to-end	96.67 %	93.33%	100%	96.67%

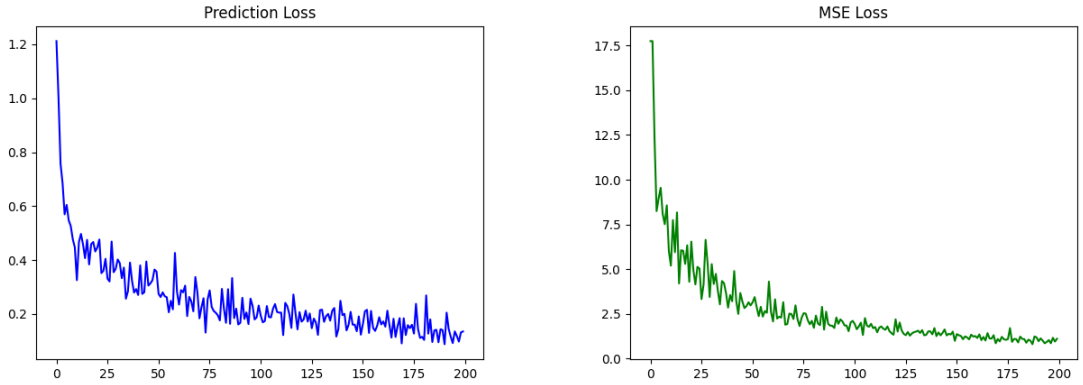
Table 3.3: Accuracy with Standard Normalization

3.2 Overall

The Ignoring and Mean filling methods performed consistently well in terms of accuracy across different percentages of missing values and normalization techniques. The Constant filling method showed a decrease in accuracy as the percentage of missing values increased, while the Network and End-to-end methods showed relatively high accuracy. Normalization techniques had an impact on the accuracy of the models, with Min Max Normalization generally showing lower accuracy compared to No Normalization and Standard Normalization.

3.3 Loss Function

This graph illustrates the loss value during the training process of the network. It seems to be better if I spend more time on improving the model and time to train.



Chapter 4

Conclusion

4.1 General

Handling missing data is a critical task in real-world data analysis, as missing data can lead to biased or incomplete results. The experiment conducted in this study evaluated different filling methods and normalization techniques for dealing with missing data in a dataset. The Ignoring and Mean filling methods showed consistent accuracy across different percentages of missing values and normalization techniques. The Constant filling method showed a decrease in accuracy as the percentage of missing values increased, while the Network and End-to-end methods showed relatively high accuracy. Normalization techniques had an impact on the accuracy of the models, with Min Max Normalization generally showing lower accuracy compared to No Normalization and Standard Normalization.

4.2 Future Work

There are several potential areas of future work for handling missing data in real-world scenarios. These include:

1. Advanced filling methods: The experiment in this study used basic filling methods such as mean and constant filling. Future work could explore more advanced filling methods, such as multiple imputation, regression imputation, or machine learning-based imputation techniques, which may provide more accurate results.

2. Ensemble methods: Combining multiple filling methods or normalization techniques through ensemble methods may potentially improve the accuracy and robustness of handling missing data. Future work could explore ensemble techniques to leverage the strengths of different methods and mitigate their weaknesses. [3]

3. Domain-specific considerations: Different domains may have specific characteristics and requirements when it comes to handling missing data. Future work could investigate domain-specific approaches for handling missing data in fields such as healthcare, finance, or social sciences, where missing data are prevalent.

References

[1] AI VIETNAM Dr Dinh Vinh Quang. End-to-end data filling network. 2023.

[2] Kostas Tzoumpas, Aaron Estrada, Pietro Miraglio, and Pietro Zambelli. A data filling methodology for time series based on cnn and (bi) lstm neural networks. *arXiv preprint arXiv:2204.09994*, 2022.

[3] Deandra Aulia Rusdah and Hendri Murfi. Xgboost in handling missing values for life insurance risk prediction. *SN Applied Sciences*, 2:1–10, 2020.