MCAR Test

IT - 462 Exploratory Data Analysis

Dhirubhai Ambani Institute of Information and Communication Technology

Group - 17

Group Members:

202312070 - Bhavish Hiranandani

202201309 - Yash Mehta

202411012 - Manish Prajapati

Introduction

Handling missing data is a frequent challenge in data analysis and machine learning. The missingpy library in Python offers a collection of tools designed to effectively address this issue. This report explores the capabilities of the missingpy library, highlighting its imputation techniques and approaches for identifying the underlying mechanisms behind missing data.

The `missingpy` library

provides practical solutions for handling missing data in machine learning, offering several imputation techniques to address incomplete datasets. These techniques can improve model performance by filling in gaps in the data.

Key Imputation Technique

MissForest: This technique uses a Random Forest model for imputation. It is capable of handling both categorical and continuous data and excels at capturing complex relationships between variables.

How it works:

- Variables are ordered by the proportion of missing values.
- Missing values are initially filled with simple estimates.
- For each variable:
    1. A Random Forest model is trained using the observed values of the variable as the target and the other variables as predictors.
    2. The model predicts the missing values.
- The process is repeated until a stopping criterion (convergence or max iterations) is reached.

```
from missingpy import MissForest

imputer = MissForest()

filled_data = imputer.fit_transform(df)
```

Understanding Missing Data Mechanisms

To effectively handle missing data, it's important to recognize the underlying mechanism behind it. Missing data can typically fall into one of three categories:

1. **MCAR (Missing Completely at Random):** In this case, the occurrence of missing data is entirely unrelated to both the observed and unobserved data. The likelihood of missingness is the same across all observations.
   Example: A machine malfunction causes a random failure to record blood pressure readings.
2. **MAR (Missing at Random):** Here, the missingness is related only to the observed data. The likelihood of a value being missing can be predicted based on other available information in the dataset.
   Example: Older individuals may be more willing to report their income, so age (an observed variable) helps explain the pattern of missing income data.
3. **MNAR (Missing Not at Random):** In this scenario, the missingness is directly related to the unobserved data itself. The reason a value is missing is inherently tied to the value that is missing.
   Example: People with higher incomes may be less likely to disclose their income, meaning the missingness is influenced by the actual income values.

Understanding these categories is essential for selecting the appropriate strategy to address missing data.

Detecting MCAR with Little’s MCAR Test

To determine whether data is Missing Completely at Random (MCAR), statisticians frequently use **Little's MCAR test**, which is available in the `missingpy` library.

Little's MCAR Test:

- **Null Hypothesis:** The data is Missing Completely at Random (MCAR).
- **Alternative Hypothesis:** The data is not MCAR.

The test generates both a test statistic and a p-value:

- If the p-value is below a selected significance threshold (typically 0.05), we reject the null hypothesis, suggesting the data is not MCAR.
- If the p-value exceeds the threshold, we fail to reject the null hypothesis, meaning the data might indeed be MCAR.

Key Steps:

1. **ınput Validation:** The function verifies that the input DataFrame is neither empty nor entirely composed of missing values, ensuring it works with valid data.
2. **Column Means Calculation:** It computes the mean of non-missing values for each column, which serves as a reference for further analysis.
3. **Missing Data Identification:** A binary mask is generated to flag missing values, helping to identify patterns of missingness across the dataset.

By performing these steps, Little's MCAR test helps evaluate whether missing data is truly random or influenced by other factors.

Implications for Data Analysis

Recognizing the missing data mechanism is crucial for choosing the right strategy to address it:

- **MCAR (Missing Completely at Random):** When data is MCAR, most imputation techniques can be applied without introducing bias into the analysis.

- **MAR (Missing at Random):** If the data is MAR, imputation methods that leverage information from other observed variables, such as multiple imputation, are recommended.
- **MNAR (Missing Not at Random):** Handling MNAR data is more complex, as the missingness depends on unobserved data. This requires explicitly modeling the mechanism behind the missing data, which can be quite difficult.

Understanding these distinctions ensures that the handling of missing data is appropriate and minimizes potential bias in the analysis.

GitHub Link for `missingpy` assignment:

https://github.com/ManishPraa24/Exploratory-Data-Analysis/tree/main/Missingpy