IT–462

Exploratory Data Analysis

Assignment-2: MCAR Test

Jash Shah (202201016)

Vraj Gandhi (202201425)

Yatri Rakholia (202411056)

Introduction:

- Handling missing data is an essential part of data analysis. Missing data generally falls into three categories:
  1. **Missing Completely at Random (MCAR):** No pattern in the missing data. The missingness is independent of any variable in the dataset.
  2. **Missing at Random (MAR):** Missing data is systematically related to other observed data, but not the missing data itself. For example, high-income individuals may be less likely to report their income, making missingness related to other variables like income.
  3. **Not Missing at Random (NMAR):** Missing data is related to the value itself. For example, people with lower income may choose not to disclose their income. NMAR requires domain expertise to confirm and cannot always be inferred from the data itself.
- Missing Completely at Random (MCAR) means that the probability of missing data on a variable is independent of any other observed or missing data in the dataset. In other words, the missingness does not depend on any observed values.
- We can check whether the missing data is of MCAR type by performing Little's MCAR Test.

Little's MCAR Test:

It is a statistical test used to assess whether the missing data in a dataset are missing completely at random or if there is a systematic pattern to the missingness. The test is named after ***`Donald R. Little`***, who introduced it.

- The MCAR assumption is an important assumption in statistical analyses, particularly in techniques like multiple imputation.
- MCAR means that the probability of missingness is unrelated to the observed or unobserved data, and there are no systematic differences between missing and observed data.

In simple terms, Little's MCAR Test is a statistical test that checks whether the missing data pattern is consistent with the MCAR assumption. It does so by:

- **Null Hypothesis (H0):** The missing data are completely at random.
- **Alternative Hypothesis (H1):** The missing data are not completely at random; there is some systematic pattern or relationship with the observed data.
- Dividing the data into groups of observed and missing values.
- Comparing the means and covariances of the observed data with expected patterns under MCAR.
- Using a **chi-squared test** to determine if the observed and expected patterns are significantly different.

- A **high p-value** indicates that missing data is likely MCAR, while a **low p-value** suggests a structured missingness (e.g., MAR or MNAR).

Implementation of Little's Test:

1) Using Python's `missingpy` Library:

- What is `missingpy`?
  The `missingpy` library is a Python package designed to handle missing data in machine learning models. It provides robust imputation methods, such as `k-Nearest Neighbors (KNN) imputation` and `MissForest`, which use observed data to estimate missing values. `missingpy` is particularly useful for datasets where missingness is assumed to be Missing at Random (MAR) or Missing Completely at Random (MCAR). Its key features are:

  ◆ **k-Nearest Neighbors (KNN) imputation**

  - **Basic Concept:** KNN is a non-parametric and instance-based algorithm, meaning it doesn't assume any underlying data distribution and directly uses the training data to make predictions.For a given data point, KNN finds the k closest data points (neighbors) from the training set based on a similarity metric (usually Euclidean distance).
  - In classification tasks, the algorithm assigns the label of a new data point by looking at the majority class among its k nearest neighbors. (i.e. **`If 3 out of 5 nearest neighbors of a data point belong to class 'A' and 2 belong to class 'B',the data point is classified as class'A'.`**)
  - In regression, instead of using the majority class, KNN predicts the target value as the average (or weighted average) of the values of its k nearest neighbors. (i.e. **`If the 5 nearest neighbors have values of 10, 15, 12, 20, and 18, the predicted value would be the mean of these values.`**)
  - The parameter k defines the number of neighbors to consider. Small k can lead to overfitting, as the model may become too sensitive to noise in the training data whereas Large k can lead to underfitting, as it averages too many neighbors, potentially including less relevant ones.
  - The choice of k is crucial and is often determined through cross-validation.
  - It relies on calculating the distance matrix between data-points. Mainly used matrices are Euclidian, Manhattan and Minkowski. For categorical data Hamming distance is used.
  - KNN algorithm is widely used in pattern recognition, image classification, document categorization and also in regression analysis

  ◆ **MissForest Imputation**

  - **Basic Concept:** MissForest is a non-parametric imputation technique based on the Random Forest algorithm, which iteratively predicts missing values

using other features in the dataset. It works for both continuous (numeric) and categorical (discrete) data, making it a versatile tool for handling missing data.

- **Working:**
  - Initially, missing values are replaced with some rough guesses (typically, the mean for continuous variables and the mode for categorical variables).
  - A Random Forest model is trained to predict the missing values in each feature. For each feature with missing values, the non-missing entries in the same feature are used as the target, while all other features are used as predictors.
  - The trained Random Forest model predicts the missing values for the feature, and these predictions are used to replace the rough guesses.
  - After all missing values are initially imputed, the process is repeated iteratively.
  - This iterative approach ensures that imputation is refined over multiple rounds, improving the quality of the imputed values.

- MissForest handles mixed data types:
  - For continuous variables, It minimizes Mean Squared Error (MSE) between predicted and actual values.
  - For categorical variables, It minimizes classification error by using the majority class predicted by the Random Forest.

- **Advantages:**
  - Since Random Forests are tree-based models, MissForest can capture complex relationships between variables, making it suitable for datasets with non-linearities and interactions between features.
  - Unlike simpler methods like mean imputation or regression-based methods, MissForest does not assume any specific distribution of the data.
  - It can handle high-dimensional datasets (many features), making it applicable to a variety of real-world use cases.

- `is_missing_mcar` function is used to perform Little's MCAR test.
- Nowadays, due to Lack of Active Maintenance and Availability of Better and Specialized Libraries `missingpy` is not used that often.
- **Improved Alternatives:** Libraries like `scikit-learn` now offer built-in imputation methods such as `KNNImputer` and `IterativeImputer`, which provide similar functionality without the need for an external library.

2) Using Manual Implementation:

Since the use of `missingpy` is declining we were not able to use it directly. Hence, we have implemented the Little's MCAR Test manually on the dataset on which we had performed the `missingno` library. You can see the user defined function in the image attached below:

```python
def little_mcar_test(data):
  n = len(data)
  groups = []
  for col in data.columns:
    mask = data[col].isnull()
    if mask.any():
      groups.append(mask.astype(int).values.reshape(-1, 1))
  if len(groups) == 0:
    raise ValueError("No missing data found.")
  r = np.concatenate(groups, axis=1)
  group_stats = r.T @ r
  m = len(groups)
  df = (n - 1) * m
  chi2_stat = group_stats.trace()
  p_value = chi2.sf(chi2_stat, df)
  return chi2_stat, df, p_value

data = pd.read_csv('/content/House Prices.csv')
chi2_stat,df,p_val = little_mcar_test(data)
print(f"Chi-square statistic: {chi2_stat}")
print(f"Degrees of freedom: {df}")
print(f"P-value: {p_val}")

if p_val < 0.05:
  print("The data are not MCAR.")
else:
  print("The data are MCAR.")
```

- **Key Components of the Function:**
  - **Detecting Missing Data:** The code loops through all columns in the dataset to identify where missing values exist. Each column's missing data is represented as a binary mask (1 for missing, 0 for observed).
  - **Combining Missingness Patterns**: All missing data patterns across columns are concatenated into a matrix `r`. This matrix shows how missingness is distributed across different columns for each observation (row).
  - **Group-level Statistics**: The function calculates co-occurrence patterns of missingness across columns. This gives insight into how often missing values in one column coincide with missing values in another column.
  - **Chi-squared Test Statistic**: The chi-squared statistic (`chi2_stat`) is calculated by summing the diagonal elements of the missingness pattern matrix (trace), which gives a simplified measure of overall missingness.

- **The degrees of freedom:** The degrees of freedom (`df`) are computed as `(n-1)*m`, where `n` is the number of observations and `m` is the number of columns with missing data. This is a heuristic for how much "freedom" the data has to vary under the null hypothesis of MCAR.
- **P-value Calculation**: The chi-squared survival function is used to compute the p-value. If the p-value is large, the null hypothesis (MCAR) is not rejected, meaning the data is likely MCAR. If the p-value is small, the null hypothesis is rejected, indicating that the missingness is likely structured (i.e., **`MAR or NMAR`**).

- **Interpretation of Results**

  - **Chi-squared statistic:** A measure of the overall missing data pattern's deviation from randomness.
  - **Degrees of Freedom (df)**: Represents how many comparisons are being made based on the missing data pattern.
  - **P-value**: Determines whether the missing data is MCAR. A large p-value (e.g., >0.05) indicates no significant deviation from randomness, implying that the missing data can be assumed to be MCAR. A small p-value suggests structured missingness.

Conclusion:

After implementing Little's MCAR test, you can draw important insights regarding the missing data patterns in your dataset. If the test results in a high p-value (commonly above 0.05), it suggests that the missing data is likely Missing Completely at Random (MCAR), meaning there is no systematic relationship between the missingness and the observed or unobserved data. In this case, simple imputation techniques or deletion of missing values may not introduce bias into your analysis. However, if the p-value is low, it indicates that the data is not MCAR, implying a need for more sophisticated imputation strategies, as the missingness might be influenced by observed variables (MAR) or unobserved factors (MNAR). Understanding whether your data is MCAR is critical to selecting appropriate imputation methods and ensuring that your analysis remains valid and unbiased.

GitHub Link: https://github.com/jash0803/DAIICT/tree/main/SEM%205/EDA

References: https://moodle.daiict.ac.in/mod/resource/view.php?id=6157