Indian Institute of Technology Dharwad
CS214: Artificial Intelligence Laboratory

Lab 2: Data Cleaning - Handling Missing Values

Note

Two problems need to be worked out in the lab. **Problem Statement 1** is compulsory. **Problem Statement 2** is a bonus.

Problem Statement 1

A dataset on red variants of the Portuguese **Vinho Verde** wine is given. This dataset contains the values of different physicochemical tests from each sample of red wine [1]. The original goal of the dataset is to model wine quality based on physicochemical tests. The attributes of the dataset based on physicochemical tests are `fixed_acidity`, `volatile_acidity`, `citric_acid`, `residual_sugar`, `chlorides`, `free_sulfur_dioxide`, `total_sulfur_dioxide`, `density`, `pH`, `sulphates`, `alcohol`, and the last attribute is `quality`. Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). You are given two CSV files. The `winequality-red_miss.csv` is a file that contains some missing values. The `winequality_red_original.csv` is the original file without missing values.

- Make a copy of the file `winequality-red_miss.csv` as `winequality-red_miss_COPY.csv`.
- Write a Python program to perform the following using `winequality-red_miss_COPY.csv`. Missing values are interpreted as `NaN` in pandas.

Steps to Follow

1. Display the number of missing values in each attribute. Also, find the total number of missing values in the file.
2. Delete any two integer values in the attribute `fixed acidity` and replace any two integer values in the attribute `volatile acidity` with `N/A`. Recalculate the number of missing values and observe the change.
3. Change any two integer values in the attribute `volatile acidity` to `na`. Recalculate the number of missing values and observe the change. If your program cannot detect `na` as a missing value, make suitable changes in the program to rectify it. And save it as a new CSV named `winequality-red_miss-modified.csv` for reference.
4. For the file `winequality-red_miss.csv`:
   (a) Count and display the number of tuples having one, two, three, four, or up to 12 missing values. Plot a graph for the `number of missing values` vs. the `number of tuples`.
   (b) Count and display the number of tuples having equal to or more than 50% of attributes with missing values.
   (c) (a) Delete (drop) the tuples equal to or more than 50% of attributes with missing values.
   (d) (b) The target (class) attribute is `quality`. Drop the tuple missing in the target (class) attribute.
   (e) Then save cleaned datasets as `winequality-red_cleaned.csv` and `winequality-red_target_cleaned.csv`.
5. Count and display the number of missing values in each attribute. Also, find the total number of missing values in the file (after the deletion of tuples).

Experiments on Filling Missing Values

1. Replace the missing values with the median of their respective attributes. (Use df.fillna() with suitable arguments.)
2. Compute the mean, median, mode, and standard deviation for each attribute and compare them with those of the original file `winequality_red_original.csv`.
3. Compare these replaced values with the actual values present in the original file. Calculate the root mean square error (RMSE) between the original and replaced

values. (Get original values from `winequality_red_original.csv`). To calculate the **Root Mean Square Error (RMSE)**, use the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}$$

where:

- x_i is the actual value.
- $\hat{x}_i$ is the predicted value.
- n is the total number of observations.

and also using `sklearn.metrics` function `mean_squared_error`

4. Calculate the root mean square error (RMSE) between the original and replaced values. (Get original values from `winequality_red_original.csv`) in percentage using the below formula:

$$\text{RMSE\%} = \left(\frac{\text{RMSE}}{\mu}\right) \times 100$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n}|x_i|$$

where:

- n: Number of samples.
- μ: Mean of the absolute values of x_i.

5. Replace the missing values by propagating previous non-missing values in that attribute. (Use df.fillna() with suitable arguments.)

6. Replace the missing values in each attribute using the linear interpolation technique. Use df.interpolate() with suitable arguments.

Problem Statement 2

A dataset related to Indian diabetes, containing medical attributes, is provided in a CSV file. The dataset includes various features that provide insights into the conditions leading to diabetes. These features are: `pregs`, `plas`, `pres`, `skin`, `test`, `BMI`, `pedi`, `Age`, and `class`. You are given two CSV files. The `pima_indians_diabetes_miss.csv` is a file that contains some missing values. The `pima_indians_diabetes_original.csv` is the original file without missing values.

- Make a copy of the file `pima_indians_diabetes_miss.csv` as `pima_indians_diabetes_miss_COPY.csv`.
- Write a Python program to perform the following using `pima_indians_diabetes_miss_COPY.csv`. Missing values are interpreted as `NaN` in pandas.

Steps to Follow

1. Display the number of missing values in each attribute. Also, find the total number of missing values in the file.
2. Delete any two integer values in the attribute `pregs` and replace any two integer values in the attribute `pedi` by `N/A`. Recalculate the number of missing values and observe the change.
3. Change any two integer values in the attribute `pedi` to `na`. Recalculate the number of missing values and observe the change. If your program cannot detect `na` as a missing value, make suitable changes in the program to rectify it. And save it as a new CSV named `pima_indians_diabetes_missupdated.csv` for reference.
4. For the file `pima_indians_diabetes_miss.csv`:
   (a) Count and display the number of tuples having one, two, three, four, or up to 12 missing values. Plot a graph for the `number of missing values` vs. `number of tuples`.
   (b) Count and display the number of tuples having equal to or more than 50% of attributes with missing values.
   (c) (a) Delete (drop) the tuples equal to or more than 50%
   (d) (b) The target attribute is `class`. Drop the tuple missing in the target `class` attribute.
   (e) Then save cleaned datasets as `pima_indians_diabetes_cleaned.csv` and `pima_indians_diabetes_target_cleaned.csv`.
5. Count and display the number of missing values in each attribute. Also, find the total number of missing values in the file (after the deletion of tuples).

Experiments on Filling Missing Values

1. Replace the missing values with the median of their respective attributes. (Use `df.fillna()` with suitable arguments.)
2. Compute the mean, median, mode, and standard deviation for each attribute and compare them with those of the original file `pima_indians_diabetes_original.csv`.

3. Compare these replaced values with the actual values present in the original file. Calculate the root mean square error (RMSE) between the original and replaced values. (Get original values from `pima_indians_diabetes_original.csv`). To calculate the **Root Mean Square Error (RMSE)**, use the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}$$

where:

- x_i is the actual value.
- $\hat{x}_i$ is the predicted value.
- n is the total number of observations.

and also using `sklearn.metrics` function `mean_squared_error`

4. Calculate the root mean square error (RMSE) between the original and replaced values. (Get original values from `winequality_red_original.csv`) in percentage using the below formula:

$$\text{RMSE\%} = \left(\frac{\text{RMSE}}{\mu}\right) \times 100$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n}|x_i|$$

where:

- n: Number of samples.
- μ: Mean of the absolute values of x_i.

5. Replace the missing values by propagating previous non-missing values in that attribute. (Use `df.fillna()` with suitable arguments.)

6. Replace the missing values in each attribute using the linear interpolation technique. Use `df.interpolate()` with suitable arguments.

Note

Please upload the completed Jupyter Notebook for each problem statement separately to Moodle for evaluation. Save the files using the format:

1. **your_rollnumber_problem_statement1.ipynb**
2. **your_rollnumber_problem_statement2.ipynb**.

References

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.