An Imputation Method for Datasets with High Percentage of Random Missing Values

Şahin Adem
Electrical And Electronics Engineering
TOBB ETU
Ankara, Turkiye
ademsahin@etu.edu.tr

***Abstract*—In this work, I examined a different data imputation method and its effects on training and test predictions performance. I used the kaggle dataset named "House Prices - Advanced Regression Techniques".**

***Index Terms*—House pricing, regression techniques, data imputation**

I. Introduction

In this work, I compared effects of different data imputation techniques on linear regression models.

II. Problem Statement

House pricing is a common problem all over the world, usually there are many missing values in datasets. Hence missing values should be treated perfectly in order to have a good end result. There are 4 types of missing values:

1) Structurally missing data
   These values are supposed to be missing. For example, area of third room must be missing if the house has only 2 rooms.
2) MCAR(missing completely at random)
   These are completely random. If a weighing machine ran out of batteries, there would be missing values until the batteries are replaced and each missing value would be unrelated. There would be no pattern.
3) MAR(Missing at random)
   If somehow there is a relation between features of the dataset, some missing values may be this type. If weight of a person is missing, it can be predicted from age, gender, height, etc. This work aims to deal with this type of missing values.
4) NMAR(Not missing at random)
   If missing value cannot be classified under the above three, it falls in this category.

III. Related Work

There are numerous house price prediction works with linear regression in the kaggle competition notebooks.One of them is done by R. Ventura [**?**]. She approaches each column in a different way when she is dealing with missing values. Mainly, she uses means, modes, medians of other samples or uses some other features directly.

IV. Method

In this work, I concentrated on predicting missing values based on other feature values of the sample. There are 79 features that can be meaningful. For each feature, I trained a RandomForestRegressor (RFR).For each feature, I used values of the other features as independent variable. The feature itself was the target variable. For each feature I divide the dataset into two, one is the samples with that feature is not missing, the other is remaining samples. I used first part as training set, and the second part as validation set. I compared this method with the classical mean, mode filling method.

A. Dataset

I worked on the data set given in a kaggle competition, named "House Prices - Advanced Regression Techniques". Data set consists of 1460 samples for training and 140 samples for test. Each sample has 80 features. 43 of the features are categorical while remaining 37 features are numeric. There are numerous columns that include missing values. Data set can be downloaded at [2].

B. Evaluation

I used Mean Squared Error metric in evaluation of the results.

C. Preprocessing

There were too few missing values for some features. Some features even has no missing values. Only 5 features have missing values more than 10 percent, which are shown in TABLE I

TABLE I
Percentage of null values in features

Feature with high percentage of null values	Percentage of Null values
PoolQC	99.5
MiscFeature	96.3
Aley	93.8
Fence	80.7

If there are too few missing values in the dataset, Filling missing values wisely does not improve results appreciably. Hence I introduced new nan values to dataset randomly in

order to compare simple imputation method. After encoding the dataset, I created 4 new datasets with ratios of missing values 0.1, 0.2, 0.3 and 0.4. Then for each dataset, I filled missing values with two different methods. One is my method, which is explained above, the other is simple filling with mean or mode of the other samples.

V. Experimental Results

I deployed a linear regressor with Lasso regularization of alpha = 0.2, then made predictions for both training and validation datasets.

A. Training results

Training results of two methods can be seen in TABLE II

TABLE II
TRAINING MSE FOR DIFFERENT IMPUTATION METHODS

NaN ratio	Simple Imputation	New method	Enhancement
0.1 NaN ratio	826639	786487	0.05
0.2 NaN ratio	886529	849707	0.04
0.3 NaN ratio	1038809	934055	0.1
0.4 NaN ratio	1087074	1001606	0.07

B. Test results

Test results of two methods can be seen in TABLE III

TABLE III
TEST MSE FOR DIFFERENT IMPUTATION METHODS

NaN ratio	Simple Imputation	New method	Enhancement
0.1 NaN ratio	4616123	4562510	0.01
0.2 NaN ratio	5431635	5258777	0.03
0.3 NaN ratio	6132312	6517582	-0.06
0.4 NaN ratio	5106699	6095594	-0.19

VI. Discussion

A. Performance Enhancement

Comparing the training scores and test scores it is clearly understood that the models overfit the training dataset. However this is not main concern in this work. For all ratios of missing values, this new method of imputation performs better than simple imputation in terms of training MSE. New imputation method is better than simple imputation method on test dataset for missing value ratios of 0.1 and 0.2. Method is worse than simple imputation for higher missing value ratios. This is not expected.

B. Overfitting Issue

In this imputation method, prediction of missing is not optimized in terms of hyperparameters. Overfitting to the training dataset may come from this issue. Also linear regression model, which is used in target prediction, is not optimized in terms of hyperparameters. These two optimization may enhance the performance of this new imputation on the final scores.

C. Types of Missing Values

The introduced missing values are absolutely in category MAR(Missing at Random). However, The original dataset already includes a number of missing values, which may degrade the performance of this imputation.

VII. Conclusion

It seem that this method of dealing with missing values can enhance the performance of preprocessing stage of the flow. Codes and results can be reached at [3].
Model can be launched online with preprocessing at [4]

References

[1] https://www.kaggle.com/code/rbyron/simple-linear-regression-models

[2] https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

[3] https://github.com/asahin24/HousePricing

[4] https://asahin24-housepricing-streamlit-deployment-ir0rom.streamlit.app/