Dealing With Missing Values - Comparison & Analysis of Different Methods

Tzach Cohen 208930842 — Ben Azulay 330485269

March 22, 2022

Abstract

In this project, we chose to deal with the missing values (in a dataset) problem. Our project wasn't meant to suggest a new solution, but rather compare existing solutions and methods throughout different datasets, and analyze the results. We chose the solution methods of Deleting rows, replacing empty values with mean/median, assigning a unique category to the empty values and using algorithms that support missing values. The results in most cases did match our expectations and predictions, which were based on the article from which we read about the existing solutions. The rest of our findings are found in the experimental results section of this document.

Problem Description

When dealing with tabular data, sometimes we can encounter what we call missing values. Missing values/data is defined as the values or data that is not stored (or not present) for some variables in the given dataset, and can be caused due to various reasons, such as human error or errors in the data formatting/saving. Such errors can bias the results of the machine learning models and/or reduce the accuracy of the ML model.

Solution Overview

After searching for a solution, we discovered that some methods of filling the data were created to deal with this issue. We decided to test 4 of them as an experiment and compare their results, pros and cons. The 4 methods we chose were:

1. Deleting rows - Missing values can be handled by deleting the rows or columns having null values. If columns have more than half of the rows as null then the entire column can be dropped. The rows which are having one or more columns values as null can also be dropped.

    - Pros
        - A model trained with the removal of all missing values creates a robust model.
    - Cons
        - Loss of a lot of information.
        - Works poorly if the percentage of missing values is excessive in comparison to the complete dataset.

2. Replacing With Mean/Median - Columns in the dataset which are having numeric continuous values can be replaced with the mean, median, or mode of remaining values in the column. This method can prevent the loss of data compared to the earlier method.

    - Pros
        - Prevent data loss which results in deletion of rows or columns.
        - Works well with a small dataset and is easy to implement.
    - Cons
        - Works only with numerical continuous variables.
        - Can cause data leakage.
        - Do not factor the co-variance between features.

3. Assigning a unique category - If the number of missing values is very large then it can be replaced with a new category.

   - Pros
     - Prevent data loss which results in deletion of rows or columns.
     - Works well with a small dataset and is easy to implement.
     - Negates the loss of data by adding a unique category.
   - Cons
     - Works only with categorical variables.
     - Addition of new features to the model while encoding, which may result in poor performance.

4. Using algorithms that support missing values (such as knn) - The k-NN algorithm can ignore a column from a distance measure when a value is missing. Naive Bayes can also support missing values when making a prediction. These algorithms can be used when the dataset contains null or missing values.

   - Pros
     - No need to handle missing values in each column as ML algorithms will handle them efficiently.
   - Cons
     - No implementation of these ML algorithms in the scikit-learn library.

Experimental Evaluation

After reviewing the solutions, their pros and their cons, we started implementing them on 4 different datasets and compared the results to see if our assumptions of each solution would be correct, using the same generic model to test each dataset (since our goal is not to increase the accuracy as much as we can, but to evaluate each method and their effects on the prediction process). We observed that:

- In 2 out of 4 datasets, the deletion of rows caused the model to be less precise, and thus have the lowest accuracy. This was expected since we deduced that the deletion of rows is actually a loss of data and therefore the model will have less data to train on than the other models/methods.
- However, we can notice that the other 2 were reduced by a quite small amount of rows, which didn't have much effect on the accuracy. From this we deduce that row deletion method is favored in cases in which the rows including missing values are in small amounts.
- We learned in class that the alteration of the dataset needs to be done after the train-test splitting, since if we calculate it before the splitting the training might be biased. This was proven as correct after we tested it ourselves - The accuracy got lower after we changed the pipeline as mentioned earlier. (We documented the accuracy change of the house prices dataset in the appendix)
- Filling the missing values with the mean percentage have proven to be the best model with the highest accuracies in most datasets. This was also expected, since the mean percentage method is a statistical approach to handle the missing values.
- Assigning a unique category for missing values performed better than deleting the rows, but it wasn't as good as the other methods. We deduce this happens because the model has now one more category to consider at the learning process, and in most cases, very few samples of this category, so the model's learning isn't efficient.
- After a couple of tries, we chose to use the KNN algorithm with 5 closest neighbors since this number produced the best results. Further increasing of the neighbors didn't improve the model, so we stayed with 5.

- The model we chose was based on multiple examples of models from competitions. We worked with simple linear regression models that were imported from the sklearn library. Since the purpose of our work is a comparison of different solution methods, all the things except the filled datasets are constant, because in an experiment you need the test subject to be the only variable.

Related Works

In our solution to the problem, we used existing tools (sklearn) and ideas. We based our approach to the problem on the following article. Out of the 7 common methods to approach the problem, we chose 4 that interested us and were the most different from each other. Our solution did not try to innovate or change the given solutions in any way, but rather conduct an experiment in which we implement the explanations and theories given in this article into a practical code and models. Later we would compare their results on different datasets and deduce whether the assumptions written in the article are correct or not necessarily correct.

Conclusion

To summarize, we found that the solutions given in the article are mostly valid and are clearly expressed in the practical results. We chose to do our project on the subject of missing values in datasets and their importance on the overall prediction process since we didn't focus too much on this specific problem and its possible solutions on the course, and through this project we learned about many easy-to-implement solutions to this given problem. We also learned that it is not a minor issue – dealing with missing values with a certain way can cause a drop in the accuracy of the model (for example, the difference in the accuracies of the deleting rows method vs the others sometimes exceeded 10%!). We've learned that there's importance in both dealing with the missing values problem, but also how you choose your solution based on the given dataset.

Appendix

- Results and Data
  - Dataset : House Prices
    - Method : Deleting Rows (Acc: 68%, before changing the pipeline: 73%)
    - Method : Mean Average (Acc: 84%, before changing the pipeline: 85%)
    - Method : Unique Category (Acc: 84%, before changing the pipeline: 85%)
    - Method : KNN (Acc: 72%, before changing the pipeline we encountered errors)
  - Dataset : Indians Diabetes
    - Method : Deleting Rows (Acc: 78%)
    - Method : Mean Average (Acc: 76%)
    - Method : Unique Category (Acc: 75%)
    - Method : KNN (Acc: 77%)
  - Dataset : Titanic Survival
    - Method : Deleting Rows (Acc: 71%)
    - Method : Mean Average (Acc: 84%)
    - Method : Unique Category (Acc: 82%)
    - Method : KNN (Acc: 83%)
  - Dataset : Wine Types
    - Method : Deleting Rows (Acc: 98%)
    - Method : Mean Average (Acc: 98%)
    - Method : Unique Category (Acc: 94%)
    - Method : KNN (Acc: 97%)