Handling Missing Values in Machine Learning Regression Problems

Danylo Shumeiko and Iryna Rozora

Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., 01601 Kyiv, Ukraine

Abstract

The problem of missing values is prevalent in practically any field that has to deal with collecting, storing, and processing large volumes of data. There are several methods for dealing with this issue, however, it is not always clear which one is optimal in a given set of circumstances. In this study, four popular methods of handling these missing values were chosen - dropping of rows, simple mean imputation, nearest neighbor imputation and multiple imputation. These methods were tested with the goal of determining whether one performs better than others across multiple models as well as determining whether the type of model has an impact on the method's effectiveness. These methods were employed on a dataset of house sales with over 21,000 entries with the price as the prediction target. The results showed that across several models multiple imputation performed most optimally, but also the fact that the comparative effectiveness of the methods does vary depending on the type of machine learning model use.

Keywords [1]

Missing Values; Machine Learning; Imputation; Regression; Prediction models

1. Introduction

The relevance of the study comes from the prevalence of the missing data problem in most statistics tasks today. The sheer volume and amount of data that needs to be processed has grown and continues to grow exponentially and due to this a lot of values and observations are bound to be missing when they get to a data analyst, data scientist, machine learning expert or any other practitioner of statistics. Handling these blank spots in a dataset is an issue that people today struggle at all levels, from student to statistician expert. Thus, there is constantly a need for research in the field. Looking at missing data in a general sense can give insights that can be then applied to any problem that uses data, but arguably looking at the problem in a more specialized way could lead to new discoveries specific to the problem or the used method. Regression problems today account for a large chunk of statistics and data science problems with new state-of-the-art models being developed often. Therefore, looking at the issue of missing values in the context of specifically regression problems, this study would allow a more detailed comparative look at the impact of different methods on certain models.

As more models are being developed in the machine learning field for the purpose of solving regression problems the issue of missing values in a dataset is more relevant than it has ever been. Because of this, a closer comparative exploration has to be performed of popular models used for regression in order to determine the best course of action for dealing with the mentioned problem.

The paper consists of five sections. Section 2 is devoted to some methods of machine learning that will be used in practical experience, and the problem of regression models in the term of machine learning analysis. Section 3 deals with missing data issues and some approaches how can the missing values be handled. There is important problem in the context of using machine learning. Most of the machine learning algorithms that are available for use today do not accept input with empty cells in

This work has been partially supported by Ministry of Education and Science of Ukraine: Grant of the Ministry of Education and Science of Ukraine for perspective development of a scientific direction "Mathematical sciences and natural sciences" at Taras Shevchenko National University of Kyiv

II International Scientific Symposium «Intelligent Solutions» IntSol-2021, September 28–30, 2021, Kyiv-Uzhhorod, Ukraine
EMAIL: shumeykodanil2@gmail.com (A. 1); irozora@bigmir.net (A. 2);
ORCID: 000-0002-8733-7559 (A. 2);

the dataset (there are exceptions as some machine learning algorithms and libraries do allow missing data through automatic handling).To combat this issue in a manner that maximizes the accuracy of predictions numerous methods and precautions have been developed. In this section, we will be looking at some of the powerful and widespread methods of handling missing data. In section 4, the practical experiment is studied and a test is carried out in order to determine the effectiveness of several different methods of handling missing data and see how these methods compare to each other. The dataset used in this study is the "House Sales in King County, USA" public dataset [20]. The goal of the machine learning models will be to predict the price of the house based on the values of other rows. Here we have 18 features (predictor variables) and 1 outcome variable – the price. The last section is conclusion.

2. Methods of Machine Learning

2.1. Regression problems and models

The first thing that should be clarified in this work is what is a regression problem in our understanding. In the terms of the current data science field the most simple and straightforward answer to that would be that a regression problem is a problem where the value of some numerical variable needs to be predicted given a dataset of observations. And really, this is the goal of regression – given input data predict an output variable, or multiple variables. But to achieve this goal we need to solve an underlying statistical modeling problem of trying to fit a model, or function, to a set of observations so that the model is accurate enough to make predictions given new observations. Let us discuss which models will be used in this work.

2.2. General linear regression model

This is a model that is not outright used in this work, however, it is required to have an understanding of this model nonetheless as an extension of it, the polynomial regression, is used as well as several other machine learning models and imputation methods relying on this model.

First let us define notation for our regression models. Let $Y = Y_1, \dots, Y_k$ be the vector of variables we want to predict, referred to as dependent or outcome variables. Often, Y is a single scalar, but it can also consist of multiple variables in more complex problems. Let $X = X_1, \dots, X_n$ be the vector of predictor variables. In simple regression problems, X can be a single predictor variable but in terms of the problems discussed in this work an n value greater than 1 is typically the case. Then, let $\beta = \beta_0, \beta_1, \dots, \beta_n$ denote a vector of regression coefficients. Here β_0 is an optional parameter and $\beta_1, \dots, \beta_n$ are the regression coefficients where each coefficient corresponds with an X variable. Then, we can write down the weighted sum of X and β known as a linear predictor [1], [2]

$$X\beta = \beta_0 + X_1\beta_1 + \cdots + X_n\beta_n \quad (1)$$

The general formula for most regression models can be written down as a function of our X variables and β coefficient with an added error term e_i representing statistical noise or some determinants of *Y* that were left unmodeled:

$$Y = f(X, \beta) + e_i \quad (2)$$

As formulated in *[1, p14]* if a general regression model is given by *f(X)* then a general linear regression model will only involve *X* as a weighted sum of all X_i, that is, it is not a function of *X* but a function of the linear predictor *X*β. Such a model is given by *f(X*β*)*. A general or generalized regression model has both linear and logistic regression as cases (more various classes of a general linear model are demonstrated in *[2])*. As outlined in *[2, p109]* a general linear regression model involves the previously defined vectors of predictors X and coefficients β, a data vector $y = (y_1, \dots, y_k)$ and also the following parameters. Firstly, the link function *g* that yields a vector of transformed data $y' = g^{-1}(X\beta)$ which is then used in modeling of the data. Secondly, the data distribution, *p(y | y')*. And lastly, some other parameters that may or may not be needed. Such parameters may include but are not limited to variances, overdispersions and cutpoints. [1], [2]

This foundation gives us a basic model to work with. However, another aspect to consider is that when we are solving a regression problem, we may find that, as often is the case, some variables in X have an effect on each other in such capacity that they cannot be separate from one another. Let us say

we have two variables, X_i and X_j that have such an interaction between them. A rather simple way to include such a dependency or interaction (as it is referred to as in *[1])* is to create a new variable, X_iX_j and add it into our model along with a new coefficient β_{n+1} such that $X\beta$ will now be written as:

$$X\beta = \beta_0 + X_1\beta_1 + \cdots + X_n\beta_n + X_iX_j\beta_{n+1} \quad (3)$$

Thus, we now compensate for this interaction between X_i and X_j in the model, although this is just one rather simplistic method of doing so.

2.3. Polynomial regression models

Polynomial models get their name from polynomials and the need for these models arises where the relationships between the predictors and the output variables are not linear, in other terms when fitting a linear function, a straight line, would not be possible to do within reasonable accuracy. A polynomial over variables X can be written in the following form (a formulation from [10]):

$$Y = \beta_0 + \sum_{i=1}^{m} \beta_i T_i + e \quad (4)$$

Where $T_i = \prod_{j=1}^{n} X_j^{a_{i,j}}$ are the terms, $a_{i,j} > 0$ are variable degrees and β are coefficients and e is the error term.

The polynomial model is powerful since it has properties that are well understood, it is flexible and therefore can fit many shapes of data, it is easy to use computationally. These are far from the only benefits of these models and they have found widespread use for various statistical and machine learning problems.

2.4. Gradient boosting models

When data scientists use most methods of regression, they attempt to find a function F(X) that fits the data with acceptable accuracy. Understandably, as such function only approximates the data and, in most cases, cannot describe it perfectly, there is also an error term that needs to be accounted for. Sometimes, however, it is the case that the relationship between the predictors and the outcome variable is not fully defined or described. In those cases, instead of being a constant error term, the error actually corelates with Y. Therefore, a secondary model can be trained on this error term and thus we would derive a formula for the error term which consists of some function h(X) and a new error term. We can then derive the updated model function as

$$F_2(X_i) = F_1(X_i) + h_1(X_i) \quad (5)$$

We can then iterate this process for n steps until we get a suitably accurate model. Here the function h can be any weak learner such as a linear model or an artificial neural network. The error of the model, or the loss, is then minimized (using methods such as gradient descent) to find optimal values. [12]

2.5. Decision trees

Decision trees are models that function by splitting the dataset into smaller and smaller subsets using simple decision models. The final tree model will consist of leaf and decision nodes that branch off the main (root) node. A decision node is a mode of the tree where an attribute of the model is being tested (for example, is $X_3 < 4.5$?) and the node is then connected to two or more other nodes which split the set further based on the result of the attribute test. The leaf nodes are the final point of the model, they represent the final decision on the value of the output variable Y.

2.6. Ensemble learning models

Ensemble models are models which use several other sub-models that train on one dataset and then have their outputs combined. Each of the model's output is accounted into the final output of the ensemble model in a 'voting' process, at times with different weights. The main aim of such a model it to produce more accurate outputs as an ensemble, even if the outputs of individual models are less accurate. [14]. The individual models can be practically anything, including a portion of the models that were discussed in this paper. The ensemble learning model that will be used in this work is a Random Forest, a model that uses decision trees.

3. Missing Data

At the moment missing data is arguably one of the largest and most relevant issues in the field of data science. Data scientists of all experience levels, both industry veterans and beginner students have to face this problem. But why does missing data present such an issue? Well, for one, it reduces the statistical power of our dataset assuming that the rows with the missing values cannot be used. Additionally, numerous other issues such as poor representativeness of the remaining sample or biasness can arise. For these reasons, simply removing and forgetting about (also known as dropping) the rows or columns with the missing data is a poor option that would lead to inaccurate result in practically any study or data science problem.

There is also the issue in the context of using machine learning. Most of the machine learning algorithms that are available for use today do not accept input with empty cells in the dataset (there are exceptions as some machine learning algorithms and libraries do allow missing data through automatic handling). There is, of course, the option of dropping the rows or columns with the missing data, however, we will then face the considerable issues that were just described above.

To combat this issue in a manner that maximizes the accuracy of predictions numerous methods and precautions have been developed. In the following sections, we will be looking at some of the powerful and widespread methods of handling missing data.

After discussing this issue, a question arises: why does it occur? What is the root of the problem of missing data? The reasons for it are often specific to the case and have to do with the way the data is collected. In general, however, it occurs either when there is an error in recording or storing a data entry or when the missing data does not actually exist. This distinction is of the highest importance as it can make a difference in how we approach the situation. In the following sections, we will take a look at methods of imputing missing data, however, using such methods is only justified in the case that the data actually exists. If the missingness of a value is due to it's nonexistence in the real world, then a different approach is needed, such as replacing the missing values with a placeholder values, like 0, and constructing another auxiliary boolean feature that indicates whether the respective values was missing.

3.1. Types of missing data

Missing data can be classified into several groups based on several different characteristics. It is important to understand these to have a solid foundation to then make decisions on how to handle the missing data. Let us now explore some of these classifications more closely. First, missing data can be divided into two groups based on number of response (dependent, output) variables affected. If just a single response variable is affected, then the missing data is defined as univariate. If multiple variables are instead affected, then the missing data is defined as multivariable. [15]

Missing data can also be unit non-responsive when the data is missing for a whole unit, or observation or item non-responsive when the failure of obtaining data affected only one or several of the features of the observation or unit, but not all. To return to our previous example with the housing prices dataset, if a whole row was missing data, in other words we lacked all data for a given house, this instance would be considered a unit non-response, while if the data was only missing for the garage area for this particular house then it would be considered an item non-response [15]

In [Rubin, 16] missing data is classified into 3 groups based on assumptions about the data – there is data Missing at Random, Missing Completely at Random and Missing Not at Random.

To better define these classes let us use notation introduced in [15]. Let U be a finite universe of N unites and let s be a sample of size n. Then, let D denote the data matrix (complete) with the element d_{ij}(i=1,…n; j=1,…J). Let D_{obs} denote the part of the matrix D that is observed and D_{miss} the part that is missing. Then, the matrix R is made up of elements r_{ij}such that r_{ij} equals to 1 when d_{ij} is observed and equal to 0 when d_{ij} is not observed. Then, the missing data problem can be defined as the fact that the probability density function of the distribution $f(R|D)$ (also referred to as the nonresponse mechanism) is unknown. [15]. Then, we can introduce the three classifications from [16] as follows. Data missing completely at random, or MCAR can be defined as data where f(R| D_{obs}, D_{miss}) = f(R) occurs if the missingness does not depend on either D_{obs} or D_{miss}. In other words, missingness depends on the characteristics of neither the observed not the unobserved values of the dataset. In this case, in terms of any analysis that is going to be performed, missing values are not treated differently from ones that are not missing. [15], [16]

Data missing at random, or MAR can be defined by f(R| D_{obs}, D_{miss}) = f(R | D_{obs}) occurs if the distribution of the nonresponse mechanism does not depend on missing values but might depend (is implied to depend) on the observed values. In other words, missingness is randomly distributed and is ignorable when the observed values had already been accounted for. This assumption is weaker than the previous one, MCAR, however if either of the two (MAR or MCAR) is true, then we can consider the nonresponse structure (the missing data structure) to be ignorable and therefore assume unbiased results in the analysis. [15], [16]. Data missing not at random, or MNAR, is an assumption that takes place when the distribution of the nonresponse mechanism, the missing data, depends on the values both observed and when not observed. That is we cannot account for the missingness through controlling for the variables that are observed. This situation is non-ignorable and makes analysis considerably more difficult since the missing data here depends on the events that cannot be measured by the analyst or researcher when working with the data. [15], [16]

As one can see, missing data can be classified by multiple characteristics into multiple groups and therefore there is no one universal approach to dealing with it. This is why it is important for any researcher, analyst or anyone else working with data to not only have a varied arsenal of tools of handling missing data, but also an understanding of when to use each tool.

3.2. Dropping

The downsides of simply dropping the rows or columns with the missing data were already mentioned in this paper. It is not an elegant solution to the problem of missing data and in most cases it has far more negative consequences than benefits. Despite this, this method still has the benefit of being arguably the simplest when compared to other methods of processing missing values and in some situations one can still consider this method for dealing with missing data.

One example would when a variable (column) has a large percentage of data missing. In cases like this, the column may be dropped for dimensionality reduction as it would have likely not provided added accuracy to the model. There are also a lot of cases however where even if a large portion or most of the values are missing in a column it should not be dropped as the variable could still have critical importance for the observations where it is not missing. Individual observations (rows) could be dropped too if they contain a large portion of missing data. When doing so one would also run into the risk of reducing the accuracy of the model, introducing biasness etc.

To summarize dropping as a method of dealing with missing data, one should consider the numerous drawbacks of using it before applying dropping. It is challenging to define formal rules or guideline to when this method should be used or what percentage of missing values in a row or column is the acceptable threshold for dropping, as it also depends on circumstances of the problem at hand. When data is Not Missing at Random, for example, analyzing the missing values instead of dropping them would likely lead to more accurate results. In theory, if data is Missing Completely at Random, though, the deleted observations would then be in turn also random and therefore loss of important variation would not take place [17]. To make a generalized summary, though, one can say that dropping varies heavily on a case-by-case basis and, in most applicable cases, should be used purely as a last resort when a significant portion of the data is missing, and other methods are not feasible or would have greater negative consequences on the accuracy of the model. If the data is MAR or MNAR dropping has more severe drawbacks, such as introducing biasness, reducing statistical power and missing out on key insights from the data.

3.3. Imputation, Simple Imputation

Imputation is the name for a family of techniques that compute values for the missing cells and fill them in to get a complete dataset without dropping any data. Imputation methods are classified in several ways, of which we will look at two: deterministic or random imputation, and single or multiple imputation. [15], [17]. Deterministic imputation methods always produce the same, determined outputs given the same dataset and parameters. Stochastic imputation methods, as the name suggests, have an element of randomness, and therefore may produce different outputs. [15]

Single imputation is a term used for a number of techniques such as mean replacement, single regression replacement and others. In these methods, the missing value is estimated, or predicted, only once. Multiple imputation on the other hand refers to a family of imputation methods that are essentially an extension of normal single imputation where the missing values are estimated multiple different times in order to reduce biasness in the standard error and potentially improve accuracy of prediction of the missing values. [17]

Some of the simplest imputation methods impute missing values through use of the logical relations between the variables to estimate the missing values with high probability of such predictions being accurate. For example, the mean imputation technique calculates the overall mean for the variable for every missing value in the dataset. Variations of this method exist that make use of other stochastic characteristics, such as median or mode. Other variations may impute a class mean or median instead. In this case, predictor variables are used to define such classes. [15]

These simple imputation methods have the benefit of being easy to use and applicable to a wide variety of problems where missing data is a factor. However, when using these techniques, one has to consider their disadvantages, mainly the distortion of the relationships between variables and compression of distributions of the variables. [15]

3.4. Nearest Neighbor Imputation

The nearest neighbor (NN) imputation methods are deterministic donor-based methods where the donor is selected through the 'nearest neighbor' procedure by minimizing the so-called distance between the recipient and potential donors. For this, a distance metric has to be defined as a function of the auxiliary variables. The unit with the smallest distance function value that is observed is then selected as the donor and the missing values in the non-respondent are filled with the missing values from this donor for relevant variables. The imputed value is always equal to another record in the dataset or an average of a number of other records. Thus, the key beneficial aspect of the NN method is that imputed values are real, observed values from the dataset (or averages of such values) and are not constructed approximations.[19], [15]

Another advantage of NN based methods is that they have often been shown to perform better than other donor-based methods, although this depends on the selection of the distance metric.[19]

The main issue with NN methods, addressed in [19] is that some features that are irrelevant to the imputation or excessively noisy add random perturbations to the distance metric and this reduce performance considerably. Multiple ways of dealing with this issues have been proposed, such as, as mentioned previously, using multiple neighbors and averaging their values to get the imputed value for the recipient. Although there are several proposed ways of dealing with this issue, it is still an issue that anyone that chooses to use Nearest Neighbors has to consider.

3.5. Multiple Imputation

Proposed in [16], Multiple Imputation methods are an extension of what is known as Single Imputation, where imputed values are only calculated once. In multiple imputation, these values are calculated multiple times instead. The reasons for imputing the values repeatedly, as opposed to only once, are as follows. Firstly, it reduces the random component of the imputation estimator's variance. Secondly, the variance estimation of the point estimator is simpler with multiple imputation, thus resulting in a relatively simple variance estimation formula. [15]

Multiple Imputation methods work well with missing data that is either MCAR or MAR and can be particularly useful with data that is Missing at Random (MAR). Analyzing data that has been imputed using multiple imputation is a three-step procedure. Firstly, the missing data is imputed. Secondly, Independent statistical analysis is conducted on the resulting datasets (of which there are several with multiple imputation). Finally, the results are then pooled across the imputations. [17]

Multiple imputation has the advantage of being able to be used in multivariate missing data problems across different variable types. This fact combined with the ability to produce additional micro-data files that can then be used for various research makes multiple imputation one of the most practical and powerful approaches for problems where a number of analyses needs to be performed with missing values across several different numerical and categorical variables. [15]. [17]

4. Experiment – comparison of imputation methods across ML models

The aim of this experiment is to carry out a test in order to determine the effectiveness of several different methods of handling missing data and see how these methods compare to each other. The test is also to be performed on three separate models to gain further insights and determine if the model type has an impact on comparative method effectiveness.

4.1. Experiment setup and conditions

The dataset used in this study is the "House Sales in King County, USA" public dataset [20]. The goal of the machine learning models will be to predict the price of the house based on the values of other rows. Here we have 18 features (predictor variables) and 1 outcome variable – the price. Four methods of handling missing data will be used – dropping of rows with missing values, simple imputation via the mean, nearest neighbor imputation and multiple imputation. The models to be tested are a polynomial regression model of the 2nd order, a random forest ensemble learning model and a gradient boosting model. The dataset will be injected with 10% missing values. This means that in this case we assume that the data is Missing Completely at Random (MCAR). Afterwards, the dataset will be split into training and validation subsets as needed and all four methods of handling with missing values will be used resulting in four groups of data. These datasets will then be fed into each machine learning model separately, measuring the RMSE and MAE of the resulting model. Once all the RMSE and MAE values are measured, they will then be manually written down in a table for analysis.

4.2. Tools

The code for this experiment is written in python 3.7. A number of python libraries were also used. The Numpy library for used for various auxiliary functions. The Pandas library was used for storing and working with the datasets. The scikit-learn library was used for the train\test split function as well as for the polynomial and random forest models. The xgboost library was used for its XGBoost regressor model, a gradient boosting model for regression. To evaluate the methods' performance two error metrics will be used: root mean squared error (RMSE) and mean absolute error (MAE). The reason for using several metrics as opposed to one is that different ways of calculating error have varying strengths, features and shortcomings. There is no one metric that is best for all situations and thus, due to the nature of this study it has been decided that using both RMSE and MAE would lead to a more wholistic picture of the results and less mistakes in the analysis of these results.

The mean absolute error is calculated as follows

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |e_i - o_i|}{n} \tag{6}$$

Where n is the sample size, e_i are the expected values (in our case the known price values in the validation samples) and o_i are the observed (in our case predicted) values. The root mean squared error is calculated as follows

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (e_i - o_i)^2}{n}} \tag{7}$$

Where, again, n is the sample size, e_i are the expected (known) values and o_i are the observed (predicted) values.

4.3. Results

Table 3.1

RMSE values for the method\model combinations

RMSE (root mean squared error)

imputation/model	Polynomial Regression Model	Random Forest Model	XGBoost Gradient Boosting Model
Dropping	204,173.64	162,626.13	164,587.21
Simple Mean Imputation	198,360.98	164,199.44	168,550.69
Nearest Neighbor Imputation	186,717.50	162,181.68	170,828.48
Multiple Imputation	172,011.71	158,998.93	160,595.07

Table 3.2
MAE values for the method\model combinations

MAE (mean absollute error)			
imputation/model	Polynomial Regression Model	Random Forest Model	XGBoost Gradient Boosting Model
Dropping	204,173.64	162,626.13	164,587.21
Simple Mean Imputation	114,831.57	85,802.97	94,920.79
Nearest Neighbor Imputation	110,972.54	86,450.22	92,138.34
Multiple Imputation	108,525.78	82,961.17	87,345.78

4.4. Results analysis

Before any analysis on the results can be conducted it should be noted that any conclusions or insights derived from these results are made within the context of the MCAR assumption due to the missing values being injected into the dataset at random. Therefore, these results potentially do not reflect situations where other assumptions, like MAR or MCNAR hold true. This experiment was also repeated multiple times to account for randomness and showed negligible difference in results across multiple attempts.

The first thing to be addressed is the dropping method. While, as expected, showing the worst MAE score across any model, the method managed to get a lower RMSE score than some of the other methods in the Random Forest and Gradient Boosting models. This, however, does not indicate that this method is accurate or will perform better than any other method. It is important to note that since we dropped rows, data was lost. While the results might be relatively accurate compared to other methods, because both the train and validation sets were derived from the same dataset, there was no external test dataset and data was lost biasness was potentially introduced into the model and, when combined with some random variation, resulted in lower-than-expected RMSE scores in two cases.

If we take a look at the MAE table, it is apparent that the methods follow a similar hierarchy across all three models – dropping being the worst, followed by mean imputation which performed better, followed by nearest neighbor imputation and multiple imputation performing the best. The polynomial regression model performed the worst, but also showed a consistency in this hierarchy in both MAE and RMSE values. The random forest and gradient boosting model have, on the other hand, varying hierarchies between MAE and RMSE, suggesting that the model type has some effect on how these methods compare. When looking at the RMSE table, an even more varied pattern can be seen. Considering all of this, it is obvious in the case of this particular experiment that the type of model does have an impact on how well a particular method of handling missing data performs. This might be due to the models putting different weights on certain columns, reacting to outliers and different data types differently (as some of the methods can result in rational values rather than exclusively integers). The scope of this impact, however, is hard to evaluate within the scope of this work, suggesting a larger scope and more detailed study might be needed.

The final and one of the key insights that can be derived is that, independent of the model and across both the MAE and RMSE scores, the multiple imputation method resulted in the best score in every case. This demonstrates this methods effectiveness and leads to the conclusion that in the MCAR assumption it is preferrable to the other methods tested in this experiment.

5. Conclusions

In this work, the topics of regression and machine learning were discussed before exploring the missing data problem in the context of these topics. As shown, there are many tools and methods available to deal with the ever-present problem of missing data. After testing several of these methods for regression problems across different models, two main conclusions can be made. The first, is the fact that among these methods, the Multiple Imputation method performed showed by far the most accurate results within the MCAR assumption. Second, is the fact that the type of the model, with high likelihood, had an effect on how these methods compare to each other. Although one method was shown to be the best in this case, a wider study could be conducted comparing a wider range of methods on not only several types of models but also multiple models of the same type, to confirm

that the difference is truly due to the model type and not to the underlying code of one particular model. As shown, the missing data problem in statistics and data science is an issue that is so prevalent that to this day it is in need of continuous research and study.

References

[1] Frank E. Harrell, Jr. 2015 Regression modeling strategies with applications to linear models, logistic and ordinal regression and survival analysis, *Springer series in statistics*, Second edition, Springer, Cham, https://doi.org/10.1007/978-3-319-19425-7

[2] Andrew Gelman, Jennifer Hill, Data 2006Analysis Using Regression and Multilevel/Hierarchical Models, *Analytical Methods for Social Research series*, Cambridge Univeristy Press, New York, NY.

[3] Jacob Hallman 2019 A comparative study on Linear Regression and Neural Networks for estimating order quantities of powder blends, KTH Royal Institute of Technology, Stockholm, Sweden.

[4] Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves 2017 Artificial *Neural Networks: A practical course*, Springer International Publishing, Switzerland, https://doi.org/10.1007/978-3-319-43162-8.

[5] Curley C, Krause RM, Feiock R, Hawkins CV. 2019 Dealing with Missing Data: A Comparative Exploration of Approaches Using the Integrated City Sustainability Database. *Urban Affairs Review*.55(2):591-615. https://doi.org/ 10.1177/1078087417726394

[6] Yagang Zhang, 2010 New Advances in Machine Learning, *IntechOpen*, https://doi.org/10.5772/225

[7] Chong Ho Yu 2010 Exploratory data analysis in the context of data mining and resampling. International Journal of Psychological Research, https://doi.org/10.21500/20112084.819.

[8] Stuart J. Russel and Peter Norvig 1995 Artificial Intelligence. A Modern Approach, Prentice Hall, Englewood Cliffs, New Jersey.

[9] Xi Cheng, Bohdan Khomtchouk, Norman Matloff, Pete Mohanty 2019 Polynomial Regression as an Alternative to Neural Nets, URL: https://arxiv.org/abs/1806.06850

[10] Aleksandar Peckov, 2012, A machine learning approach to polynomial regression, Ljubljana, Slovenia, URL: http://kt.ijs.si/theses/phd_aleksandar_peckov.pdf

[11] Pereira, José & Basto, Mario & Ferreira-da-Silva, Amelia 2016 The Logistic Lasso and Ridge Regression in Predicting Corporate Failure. *Procedia Economics and Finance*, 39:634-641, URL: https://www.researchgate.net/publication/305396438_The_Logistic_Lasso_and_Ridge_Regression_in_Predicting_Corporate_Failure, https://doi.org/10.1016/S2212-5671(16)30310-0.

[12] Zhang, Zhongheng & Zhao, Yiming & Canes, Aran & Steinberg, Dan & Lyashevska, Olga, 2019 Predictive analytics with gradient boosting in clinical medicine, Annals of Translational Medicine, 7(7):152-152, https://doi.org/0.21037/atm.2019.03.29.

[13] Wei-Yin Loh, Classification and Regression Trees., Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1):14-23, 2011, https://doi.org/10.1002/widm.8.

[14] Gavin Brown, 2010 Ensemble Learning, volume 310 of Encyclopedia of Machine Learning, Webb, G.I., Sammut, C., Eds.; Springer: New York, NY, USA.

[15] I. Rozora, N. Rozora 2009 Application of Imputation Methods for Sampling Estimation. Proceedings of the Baltic-Nordic-Ukrainian Summer School on Survey Statistics, 23-27 August.2009. Kyiv, "TBiMC".p.139-146.

[16] Roderick J. A. Little, Donald B. Rubin. 2019 *Statistical Analysis with Missing Data*, 3rd Edition, Wiley, 2019.

[17] Rebecca R. Andridge, and Roderick J A Little, A Review of Hot Deck Imputation for Survey Non-response, International statistical review = Revue internationale de statistique vol. 78,1 40-64, 2010, https://doi.org/10.1111/j.1751-5823.2010.00103.x

[18] Lorenzo Beretta, Alessandro Santaniello, 2016, Nearest neighbor imputation algorithms: a critical evaluation, BMC Med Inform Decis Mak 16, article number 74, https://doi.org/10.1186/s12911-016-0318-z

[19] House Sales in King Country, USA, public domain dataset, Kaggle, URL: https://www.kaggle.com/harlfoxem/housesalesprediction

[20] Snytyuk V.Ye. 2006 An evolutionary method for recovering missing data. Proceedings of VI Int.Conference "Intellectial analysis of information", Kyiv, pp.262-271.