Missing values and noise in data. How crucial it can be?

Vladislav Molodtsov [1] **Irina Shushpannikova** [1] **Stepan Vasilev** [1] **Kelvin Kutsukutsa** [1] **Zhadyraiym Akunova** [1]

Abstract

In real industrial projects, it is common that the data provided contains noise, outliers, and missing values. In this paper, by performing computational experiments with plenty benchmark datasets, we investigate how various distortions affect the quality of ML models. Moreover, we compare popular imputation methods in different scenarios of missing values in absence and presence of noise and outliers. Taking into account obtained results, we give practical recommendations on handling corrupted datasets.

Github repo: link
Video presentation: link

1. Introduction

Successful solution of any Machine Learning (ML) task extremely depends on the quality of the input data. As a rule, data is rarely perfect, especially in real-world business and industry: it is often corrupted by noise, outliers, and missing values. It is customary to use simple methods of data reduction, replacement by mean or median, and so on in order to handle this imperfection. However, such methods may lead to information loss (Mani et al., 2005). To alleviate this problem, two main directions are being developed. On the one hand, there is missing data toleration. For example, researchers in paper (Saar-Tsechansky & Provost, 2007) defined features imputability and feature part-reduction as a direct correlate to the models' effectiveness; paper (Zhu et al., 2012) is focused on dealing with missing values on train and test cases of classification dataset; paper (Hathaway & Bezdek, 2002) presents some approaches for completing distance matrices for clustering tasks; paper (Barron & Barron, 1998) discusses an application of the Markov blanket learning method on synthetic and real-world data with different level of features incompletion. On the other hand, there are missing data imputation techniques. Among the works considering this technique can be mentioned: paper (Barron & Barron, 2007), where imputation method aims at making an optimal evaluation about Root Mean Square Error (RMSE), distribution function and quantile, and papers (Little & Rubin, 2002; Tsikriktsis, 2005) investigating model- and likelihood-based imputation methods.

However, mentioned works do not consider the impact of the noise, which is inherent in real-world data. The rare exception is paper (Zhu et al., 2012) where the authors evaluate the performance of several common imputation methods in presence of noise. For measuring imputation quality, they adopted Normalized Mean Absolute Error (NMAE) between the imputed and initial datasets. But they do not consider how the imputation methods affect different ML models, especially in presence of noise. Thus, our work aims to expand the study in paper (Zhu et al., 2012) and investigate the impact of noise, outliers, and missing values on the quality of various machine learning models by making computational experiments. We take 5 tabular benchmark datasets for both regression and classification problems, simulate data corruption in various ways and with different corruption levels, and test different imputation techniques and ML models. Based on our findings, we are going to give some practical recommendations on how to handle missing values and which models are the most robust to noise.

The paper organization is the following. In Section 2, we describe i) used datasets; ii) metrics used for evaluating the quality of ML models and imputation methods; iii) scenarios of adding noise and missing values; iv) imputation methods to be compared. In Section 3, we present several pipelines of computational experiments, as well as the obtained results. Section 4 concludes the paper.

The main contributions of this report are as follows. First, we investigate how vulnerable ML models are to the different kinds of noise in datasets. Second, we compare several imputation techniques on real-world datasets considering a set of missing data mechanisms. Finally, we study how imputation methods are affected by noise and give some practical recommendations.

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Vladislav Molodtsov <vladislav.molodtsov@skoltech.ru>.

Final Project on the Machine Learning 2022 Course, Skoltech, Moscow, Russian Federation, 2022.

Table 1. Datasets description

	NAME	PROBLEM	DESCRIPTION	SHAPE	TARGET
1	AIR TEMPERATURES	REGRESSION	PREDICT AIR TEMPERATURE BY EXTERNAL DATA	(7588, 23)	MIN VALUE 17.4 MAX VALUE 38.9
2	AIR QUALITY	REGRESSION	IDENTIFY AIR QUALITY BY SENSORS DATA	(827, 13)	MIN VALUE 0.4 MAX VALUE 1.5
3	PARKINSON DISEASE	REGRESSION	PREDICT PARKINSON DISEASE BY VOICE MEASUREMENTS	(5875, 22)	MIN VALUE 7 MAX VALUE 55
4	WINE QUALITY	CLASSIFICATION	IDENTIFY WINE QUALITY BY PHYSICOCHEMICAL TESTS	(4898, 12)	7 CLASSES, FROM 5 TO 2198 ELEMENTS
5	ROBOT'S SENSORS	CLASSIFICATION	PREDICT ACTION BY SENSORS DATA	(5455, 25)	4 CLASSES, FROM 328 TO 2205 ELEMENTS

2. Preliminary

In this section, we describe preliminary things needed for full understanding of the carried out experiments. In Section 2.1, we describe datasets being used. Section 2.2 explains how we evaluate the quality of ML models and imputation methods. In Sections 2.3 and 2.4, we represent the considered scenarios of adding noise and missing values, respectively. Finally, in Section 2.4, the description of used imputation techniques is given.

2.1. Datasets description

For our computational experiments, we take several tabular benchmark datasets from archive.ics.uci.edu. The datasets originate from different domains, contain both continuous and categorical features, and are devoted to both regression and classification problems. Their descriptions as well as the links for downloading are represented in Table 1. Note that in case of regression, the scale of the target is quite different. In the first dataset, its maximal value is 38, while in the second one, it is 1.5. Also note that classification datasets are highly imbalanced. For instance, in the fourth dataset the major class has more than 2000 samples, while the minor has only 5.

The datasets were preprocessed before the experiments by dropping all samples containing missing values in order to totally control the generation of missing values in the datasets. It it important since we are going to produce missing data with some specified patterns and analyze the influence of it. In addition, all considered ML models used Standard Scaler as the first stage of the ML pipeline, since it is a common practice to normalize the data. No cleaning from outliers and anomalies was used in the work, however, their influence is eliminated by considering the introduced distortion metrics described in Section 2.2.

2.2. Used metrics

As a metric for assessing models' quality, we compute cross-validation (CV) score. We use 3-fold CV through the whole paper, thus, we do not use train-test split. For classification we use F1-micro score, as it takes into account both precision and recall, while micro averaging is preferable for imbalanced datasets as ours. For regression, we use Mean Absolute Percentage Error (MAPE)[1], as it is not sensitive to the target scale which is different in our datasets. These two metrics were used as scoring for finding optimal hyperparameters in Grid Search CV. The grid of parameters in CV and the list of considered models are represented in Section 3.1.

In sake of eliminating the influence of initial noise and outliers in the datasets, we introduce the normalized distortion metrics D_{clf} and D_{reg} while assessing the quality of imputation techniques in case of classification and regression problems, respectively. These metrics are defined as the ratio between models' quality metrics on initial and distorted datasets according to eq. (1). The meaning of these ratios is how much the dataset's quality has changed.

$$D_{clf} = \frac{F1_{distorted}}{F1_{initial}}, \quad D_{reg} = \frac{MAPE_{initial}}{MAPE_{distorted}} \tag{1}$$

We want to make these metrics grow with the increase in the imputation method's quality. In case of classification, higher F1-score is better, therefore, model's quality metric on distorted dataset $F1_{distorted}$ is in the numerator. In case of regression, lower MAPE is better and model's quality metric on distorted dataset $MAPE_{distorted}$ is in the denominator. Thus, the higher these distortion metrics, the better imputation technique works.

Note, however, that in experiments with noise only, we still consider the models' quality metrics, not the distortion metrics. It leads to the fact that the results can be significantly affected by initial noise and outliers in the dataset. Thus, we have to analyze metrics changes in presence of noise, not the absolute values.

[1]To be more precise, we use negative MAPE as scoring, since the lower MAPE, the better quality

2.3. Adding noise

For adding noise, we consider two different scenarios. The first one is adding Additive White Gaussian (AWGN) noise with a predified Signal-to-Noise Ratio (SNR). This scenario adds noise only to continuous features and leaves categorical ones as is. Implementation is the following. We iterate over only continuous features and for each of them, we compute the average square of feature which has the meaning of "signal power". Then, we add noise with the power corresponding to the defined SNR. In our experiments, we consider $SNR = \{-20dB,\ -18dB, \dots, 20dB\}$.

The second scenario is to change a value of each sample with a predefined probability $p_{ch} = \{0,\ 0.025, \dots, 0.5\}$. This approach, denoted here and below as Random Changing (RC), models outliers in the dataset and can be applied to both categorical and continuous features. Thus, we iterate over all features and change its value to another class with predefined probability p_{ch} in case of categorical feature and to some randomly selected value from min to max in case of continuous one. The usage of RC is inspired by paper (Zhu et al., 2012) where it was used as a main approach for modeling noise in the datasets.

2.4. Adding missing values

Inspired by paper (Zhu et al., 2012), we consider 3 scenarios for introducing missing values as well. In the following equations, $Pr(X|Y)$ means conditional probability of X given Y, R denotes the indicator matrix with the same size as the considered dataset, R_{ij} is equal to 1 if the corresponding element in the dataset is missed and 0 otherwise, D^m and D^o denote a set of elements being missed and observed in the initial dataset, respectively. Given such notations, implemented scenarios for adding missing values can be formulated as follows:

- *Missing Completely At Random (MCAR)*
  $P(R|D^m, D^o) = P(R)$ – Probability of missing the value does not depend on any values
- *Missing At Random (MAR)*
  $P(R|D^m, D^o) = P(R|D^o)$ – Probability of missing the value depends on some other values
- *Not Missing At Random (NMAR)*
  $P(R|D^m, D^o) = P(R|D^m)$ – Probability of missing the value depends on itself

All considered scenarios have the parameter called dropping probability p defining the rate of so-called missingness. In our experiments, we consider values $p = \{0.1,\ 0.125, \dots, 0.3\}$. The approximate implementation of these scenarios are as follows. In case of MCAR, we just iterate over all rows and columns and drop each entry with predefined dropping probability p. In case of MAR, the probability of dropping the entry depends on some other one. For simulating that, we randomly split the dataset into two equal parts, then we iterate through all entries in the first part and drop them with probability $4p$ depending on the corresponding value in the second part. Finally, in case of NMAR, dropping is systematic. We iterate over all features and drop them with probability $2p$ depending on their values. For more details about scenarios implementation, we refer interested readers to paper (Zhu et al., 2012).

2.5. Imputation techniques

Regarding data imputation, we compare 5 techniques. The first 3 are the simplest ones: filling with 0, mean, and median. The next ones are more complicated: Multiple Imputation by Chained Equation (MICE)[2], when missed features are iteratively predicted by others using Random Forest, and K-Nearest Neighbor imputation (KNN)[3], when we predict missed features by mean value of K nearest neighbors. We refer interested readers to papers (Malarvizhi & Thanamani, 2012; White et al., 2011) for more details about these methods. The used parameters are the following: 5 iterations in case of MICE and 5 neighbors in case of KNN.

3. Experiments and Results

In this section, we describe the methodology of our experiments and present the obtained results. In Section 3.1, experiments on initial datasets without distortion are represented and the initial CV scores for considered models are given. Section 3.2 provides details and results for experiments with noise only, while Section 3.3 describes those for experiment with missing values only. Finally, in Section 3.4, we consider computational experiments with introducing both noise and missing values. All reproducible python source code is available at our GitHub repository. We did not use any specific computing infrastructure for our experiments, since all computations could be easily performed on simple CPU. Note that all experiments were performed 3 times with different random seeds. After that, the results were averaged, standard deviations were computed.

3.1. Experiments on initial datasets

First, we carry out experiments on initial datasets without any distortions. The scheme of the experiments is shown in Fig. 1. Here, we are interested in tuning hyperparameters for the ML models and in computing initial CV score on datasets without distortions.

The ML models used for experiments are Linear Regres-

[2]We used implementation from miceforest GitHub repository
[3]We used function *KNNImputer* from scikit-learn

Table 2. CV scores on initial datasets

	DATASET NAME	PROBLEM	METRICS NAME	CROSS-VALIDATION SCORE			
				LINEAR	DT	RF	LIGHTGBM
1	AIR TEMPERATURES	REGRESSION	MAPE	**0.04**	0.05	**0.04**	**0.04**
2	AIR QUALITY	REGRESSION	MAPE	**0.06**	0.10	0.08	**0.06**
3	PARKINSON DISEASE	REGRESSION	MAPE	**0.11**	0.12	**0.11**	**0.11**
4	WINE QUALITY	CLASSIFICATION	F1-MICRO	0.52	0.51	**0.53**	0.52
5	ROBOT'S SENSORS	CLASSIFICATION	F1-MICRO	0.68	0.98	**0.99**	**0.99**

sion[4] (Linear), Decision Tree (DT), Random Forest (RF)[5], and Gradient Boosting (GB)[6]. We use 3-Fold CV with the following grid of parameters: number of estimators in ensemble models $n_{estimators} = \{10, 20, \dots, 100\}$, maximum depth for tree-based models $max_{depth} = \{3, 5, 7\}$, inverse of regularization strength $C = \{10^{-3}, 10^{-2}, \dots, 10^{3}\}$. Once we found optimal hyperparameters for each dataset and each ML model, we use them without any changes in all further experiments.

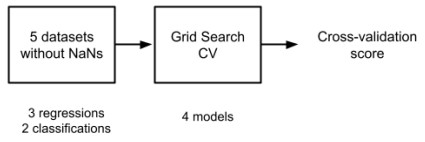

Figure 1. Pipeline of experiments with initial datasets

After fixing hyperparameters, we calculate CV score on initial datasets which are summarized in Table 2[7]. Remind that we are interested in getting MAPE as low as possible for regression and getting F1-micro as high as possible for classification. Also notice that the metrics do not differ a lot across the models.

3.2. Experiments with noise only

The next set of experiments is dataset distortion by noise. The scheme of the experiment is shown in Fig. 2. After adding noise in two different ways as discussed in Section 2.3, we again compute CV scores with hyperparameters found in previous experiment, but this time on datasets with noise. Here, our goal is to figure out how the score is affected by noise.

Note that as discussed in Section 2.1, we eliminated all missing values from the datasets, but they may contain some noise and outliers. However, as stated in Section 2.2, we are interested in metrics changes, not the absolute values, thus, the impact of initial distortions is mitigated.

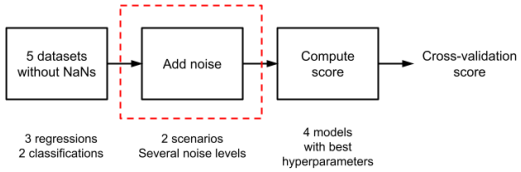

Figure 2. Pipeline of experiments with noise only

The dependencies of models' quality metrics on noise level for different considered datasets are represented in Fig. 5, 6, 7, 8, and 9 which can be found in Appendix A. The left plot is RC, the right one is AWGN. Y-axis is inverse MAPE metric in case of regression and F1-micro in case of classification. So the higher the line, the better models' quality.

From the depicted figures, we see that the lines are failing very quickly when noise increases. Thus, we can conclude that noise can significantly affect ML models' quality. Also note that ML models behave differently on various datasets in case of noise, therefore, we can not predict in advance which model is better.

3.3. Experiments with missing values only

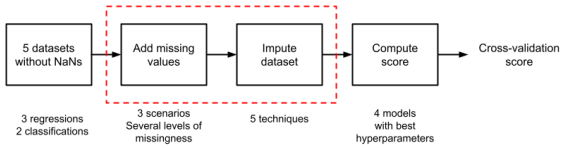

Figure 3. Pipeline of experiments with missing values only

Another experiment set is dataset distortion by missing values. The scheme of the experiment is shown in Fig. 3. After adding missing values in different ways as discussed in Section 2.4, we impute them using one of the described method from Section 2.5. Finally, we again compute CV scores with hyperparameters found before, but this time on datasets after imputation. Here, our goal is to find out how the score is affected by missing values and which imputation method is the best. For the last task, we measure distortion metrics introduced in Section 2.2.

[4]Logistic Regression is used instead in case of classification problem

[5]We used function scikit-learn library for using Linear, DT, and RF

[6]We used LightGBM from LightGBM GitHub repository, since we have huge amount of experiments and need fast model

[7]Again, by CV score for regression we mean negative MAPE, but in the table, the absolute values are given for clarity

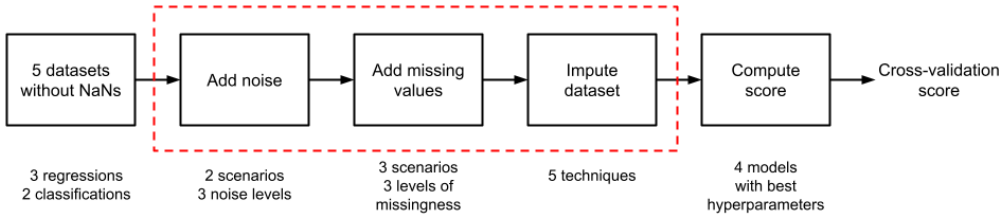

Figure 4. Pipeline of experiments with noise and missing values

The obtained results can be found in Appendix B. In Fig. 10 and 11, the dependencies of the distortion metrics on dropping probability p for different imputation methods are depicted. We consider only one dataset and one scenario of introducing missing values (MAR), since the idea is to show that the imputation methods behave similarly across all ML models. Thus, it is reasonable to average the results across ML models in order to reduce the total number of pictures.

Given averaging across the models, we can build the radar-diagram for compact results representation. Fig. 12 and 13 from in Appendix B depict such diagrams. The idea is that each colored line corresponds to some imputation method. The apexes represent the scenarios of introducing missing values. For example, MCAR(p) is MAR scenario with dropping probability p. The value of distortion metric in some missing scenario lies at the intersection of the colored line and the corresponding radius. The larger the area of the colored figure, the better the imputation method works. For instance, in case of dataset 1, MICE outperforms other imputation methods, however, in case of dataset 5, MICE fails while KNN and filling by median are the best.

Thus, we can conclude that on some datasets, MICE and KNN can perform significantly better than simple imputation methods such as 0, mean, median, but on others, MICE and KNN can fail.

3.4. Experiments with noise and missing values

In our final experiments, we analyze the effect of both noise and missing values added together. The scheme of the experiment is shown in Fig. 4. We add both noise and missing values, then impute datasets and compute the distortion metrics for every imputation method. The idea of these experiments is to inspect which imputation method is the best in the presence of noise.

For results representation, we again build radar-diagrams depicted in Fig. 14 and 15 from Appendix C. Here, we consider a regression dataset and 4 different noise levels: i) no noise; ii) AWGN of 10 dB; iii) AWGN of 0 dB; iv) AWGN of -6 dB, respectively.

From the obtained results, we can make two conclusions. First, imputation quality is significantly affected by noise. Second, when the noise level changes, other imputation methods become better, and there are no predictable patterns in this phenomenon.

4. Conclusion

In this paper, we have studied the impact of various data distortions of different levels and compared popular imputation methods in plenty scenarios. Our experiments have led us to the following conclusions. Foremost, noise and missing values can significantly affect the quality of ML models. Second, in some cases, sophisticated imputation techniques, such as MICE and KNN, are exceedingly more efficient than usually used techniques like filling by 0 or mean. However, they are not silver bullets, and in other cases, they can easily fail. Third, noise can significantly affect imputation quality. Moreover, it is unpredictable which imputation method is the best for a particular level of noise. Thus, we recommend to always try several pipelines and find the best one for each particular dataset independently by cross-validation.

References

Barron, A. and Barron, R. Statistical learning networks:a unifying view. *Wegman E (ed) Proc the 20th symposium on the interface: computing science and statistics*, pp. 192–203, 1998.

Barron, A. and Barron, R. Semi-parametric optimization for missing data imputation. *Applied Intelligence*, 27(1): 79–88, 2007.

Hathaway, R. and Bezdek, J. Clustering incomplete relational data using the non-euclidean relational fuzzy c-means algorithm. *Pattern Recogn Lett*, 23(1-3):151–160, 2002.

Little, R. and Rubin, D. *Statistical analysis with missing data*. Wiley, 2002.

Malarvizhi, R. and Thanamani, A. S. K-nearest neighbor

in missing data imputation. *International Journal of Engineering Research and Development*, 5(1):5–7, 2012.

Mani, S., Valtorta, M., and McDermott, S. Building bayesian network models in medicine: The mentor experience. *Applied Intelligence*, 22(2):93–108, 2005.

Saar-Tsechansky, M. and Provost, F. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1623–1657, 2007.

Tsikriktsis, N. A review of techniques for treating missing data in om survey research. *J Oper Manag*, 24(1):53–62, 2005.

White, I. R., Royston, P., and Wood, A. M. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

Zhu, B., He, C., and Liatsis, P. A robust missing value imputation method for noisy data. *Applied Intelligence*, 36(1):61–74, 2012.

A. Graphs for experiments with noise only

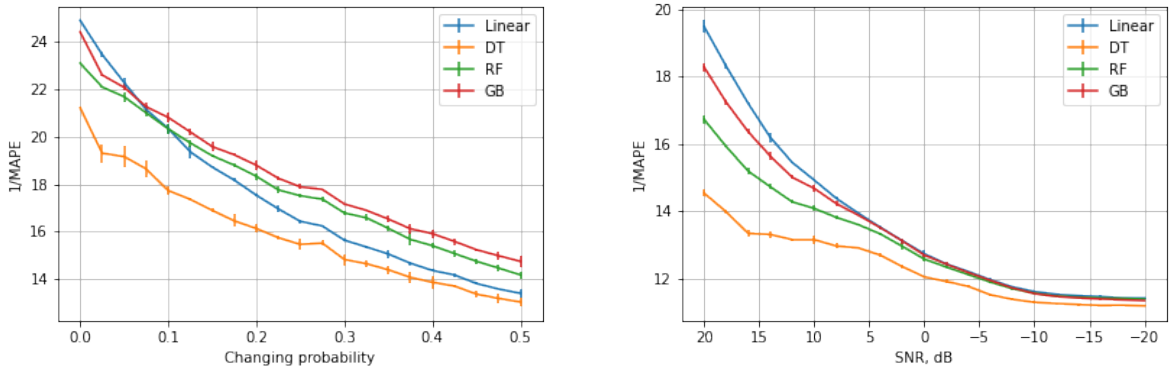

Figure 5. The dependencies of inverse MAPE on noise level for dataset 1 (Reg). Left: RC; Right: AWGN

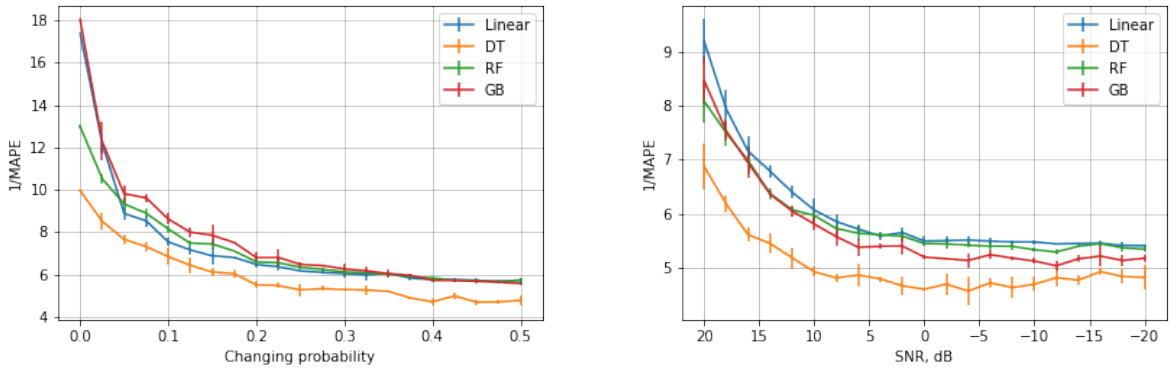

Figure 6. The dependencies of inverse MAPE on noise level for dataset 2 (Reg). Left: RC; Right: AWGN

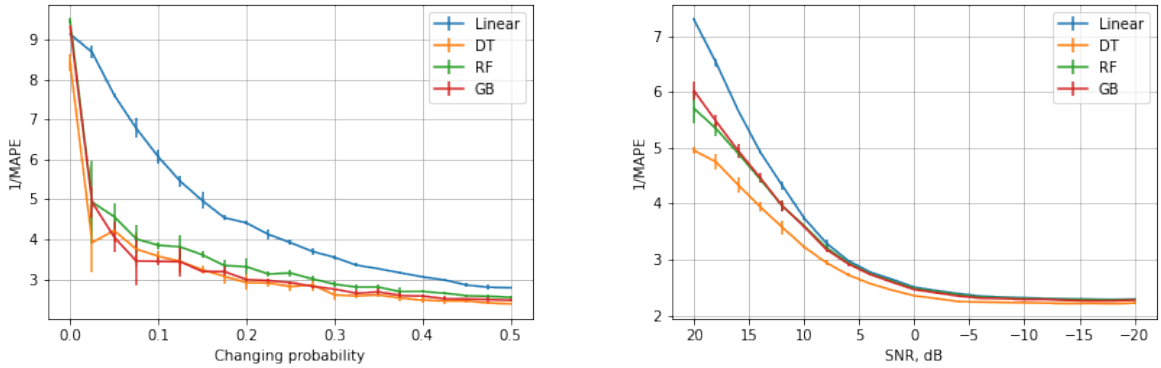

Figure 7. The dependencies of inverse MAPE on noise level for dataset 3 (Reg). Left: RC; Right: AWGN

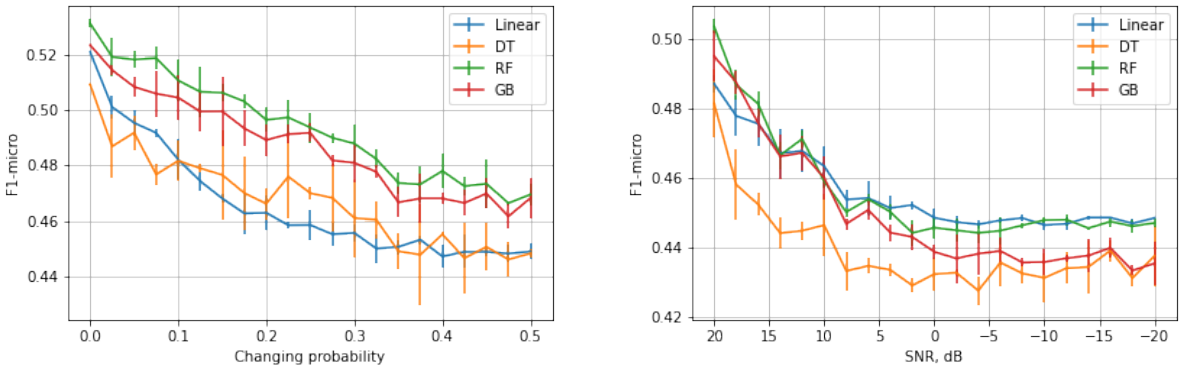

Figure 8. The dependencies of F1-micro on noise level for dataset 4 (Class). Left: RC; Right: AWGN

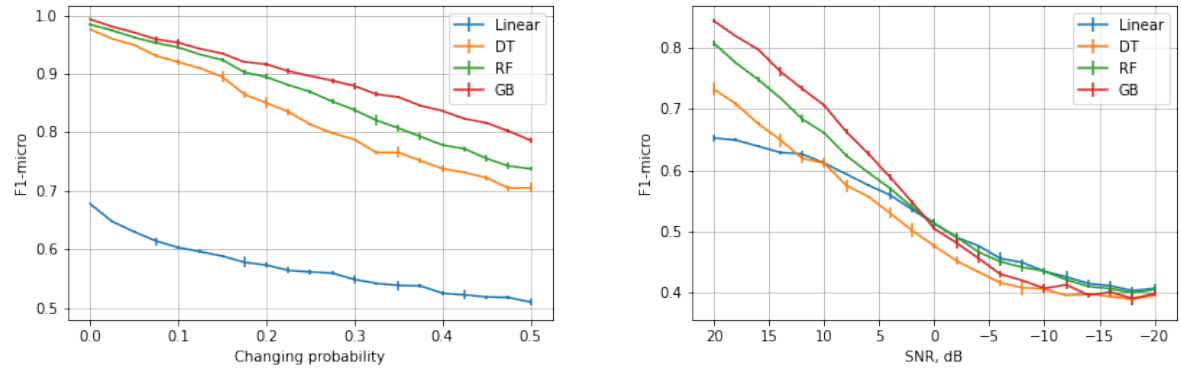

Figure 9. The dependencies of F1-micro on noise level for dataset 5 (Class).Left: RC; Right: AWGN

B. Graphs for experiments with missing values only

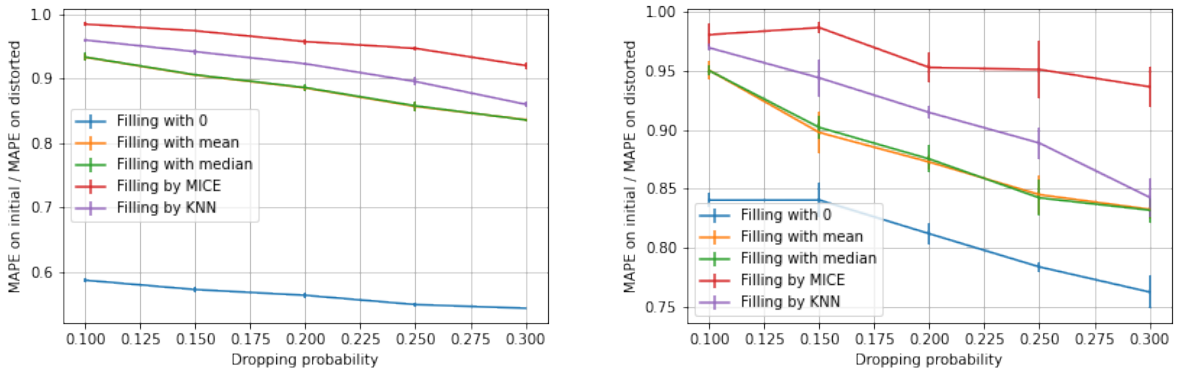

Figure 10. The dependencies of distortion metrics on dropping probability for dataset 1 (Reg). Left: Linear Model; Right: DT

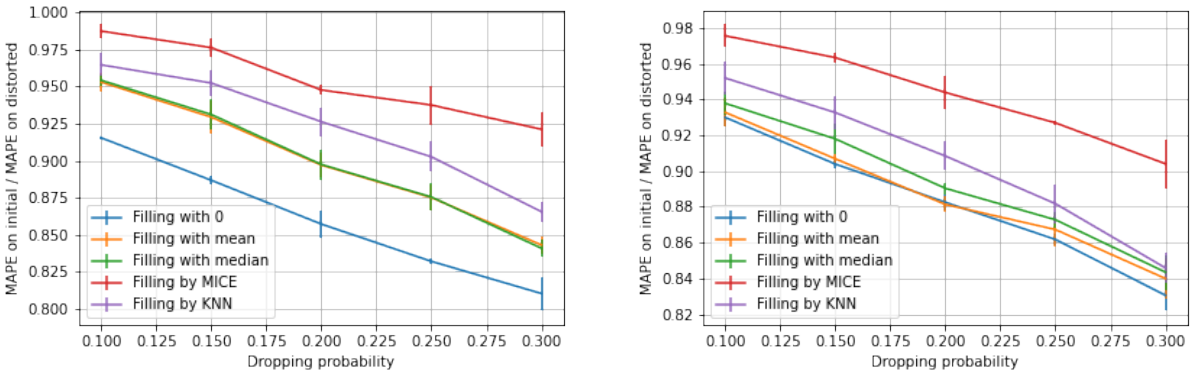

Figure 11. The dependencies of distortion metrics on dropping probability for dataset 1 (Reg). Left: RF; Right: GB

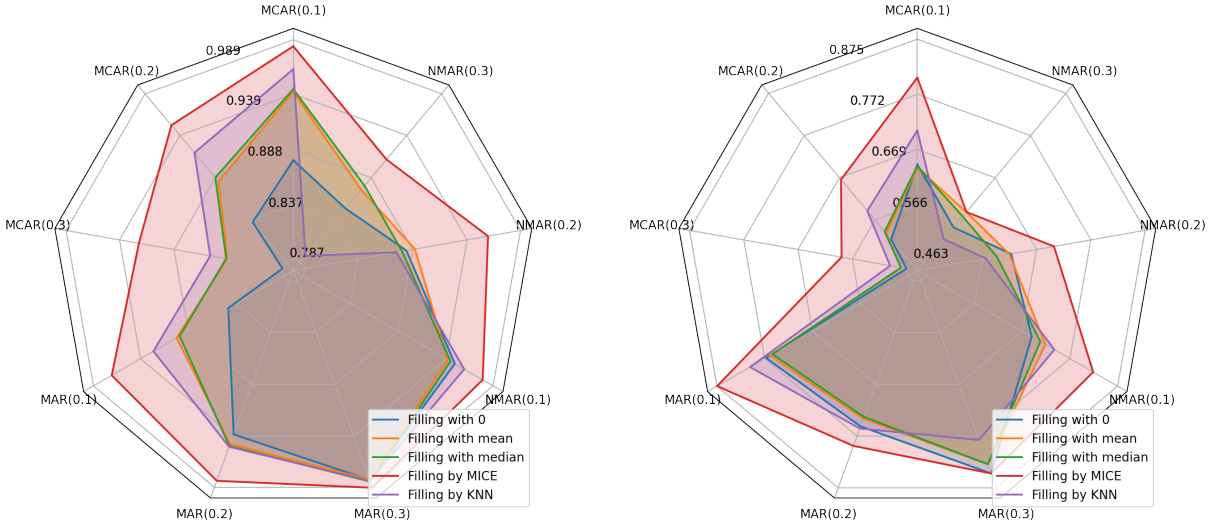

Figure 12. Radar-diagrams for regression problem without noise. Left: dataset 1. Right: dataset 2

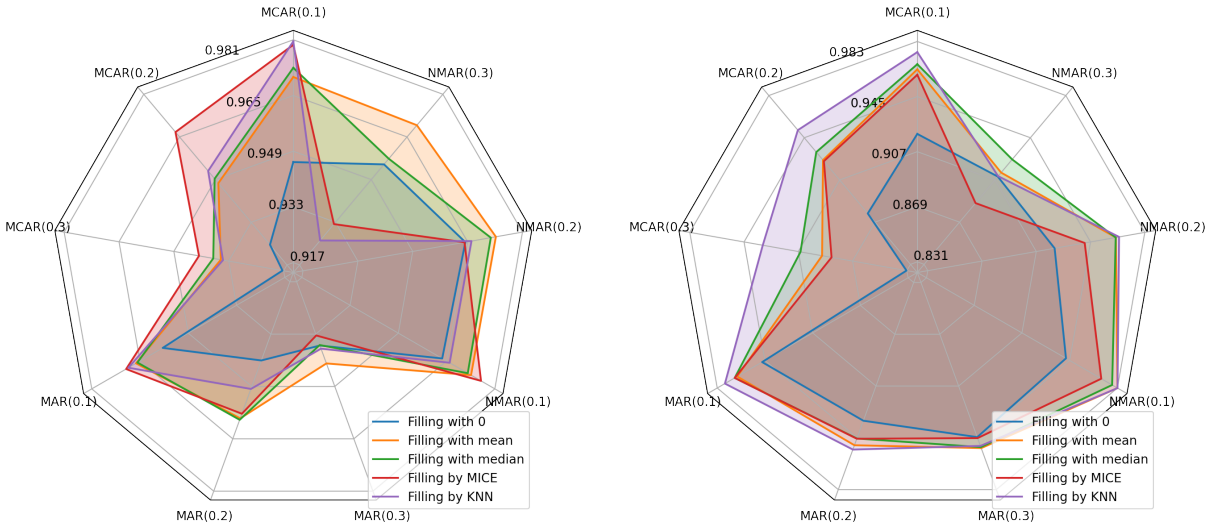

Figure 13. Radar-diagrams for classification problem without noise. Left: dataset 4. Right: dataset 5

C. Graphs for experiments with noise and missing

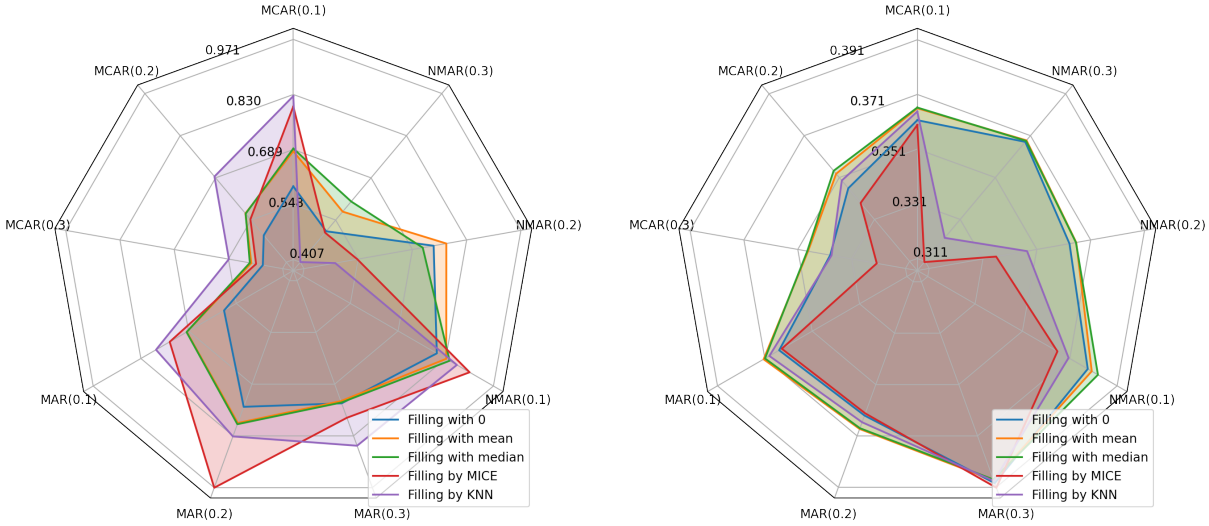

Figure 14. Radar-diagrams for different levels of noise (dataset 3, regression). Left: without noise; Right: AWGN of 10 dB SNR

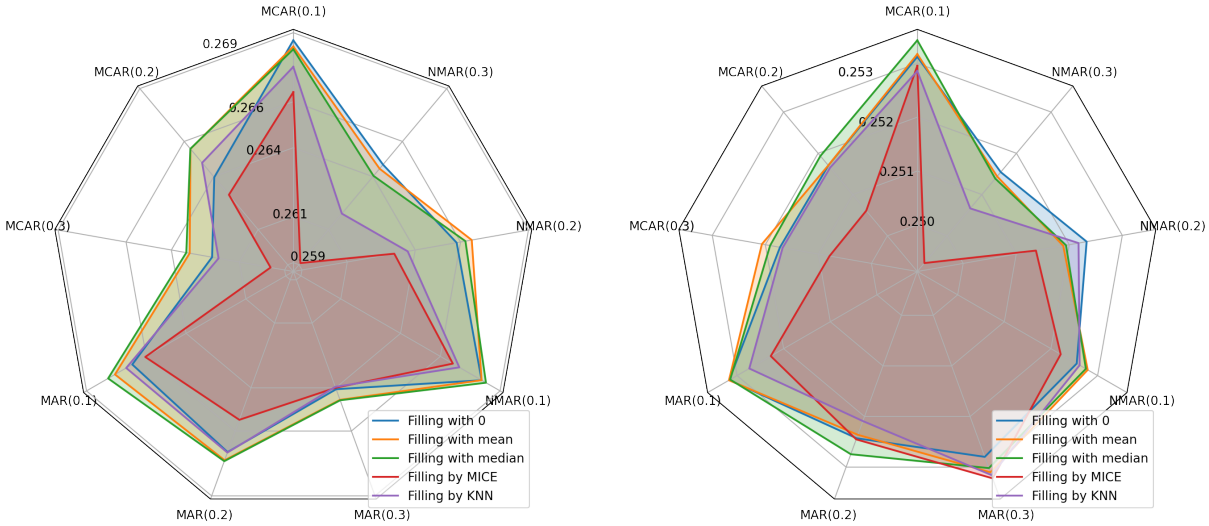

Figure 15. Radar-diagrams for different levels of noise (dataset 3, regression). Left: AWGN of 0dB SNR; Right: AWGN of -6 dB SNR

D. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Vladislav Molodtsov (30% of work)

- Leading the team and task distributing;
- Reviewing literate on the topic;
- Creating experiments methodology;
- Implementing different imputation methods;
- Merging final code;
- Troubleshooting;
- Running experiments;
- Processing results;
- Building graphs;
- Consulting with TA;
- Preparing the GitHub Repo;
- Preparing slides;
- Preparing presentation text;
- Preparing video presentation;
- Processing video;
- Preparing report.

Irina Shushpannikova (23% of work)

- Reviewing literate on the topic;
- Implementing different methods of noise adding;
- Running experiments;
- Processing results;
- Building graphs;
- Preparing slides;
- Preparing presentation text;
- Preparing video presentation;
- Preparing report.

Stepan Vasilev (26% of work)

- Reviewing literate on the topic;
- Implementing models evaluation;
- Running experiments;
- Processing results;
- Building graphs;
- Consulting with TA;
- Preparing slides;
- Preparing presentation text;
- Preparing video presentation;
- Preparing report.

Kelvin Kutsukutsa (15% of work)

- Reviewing literate on the topic;
- Implementing scenarios of adding missing values;
- Running experiments;
- Preparing slides;
- Preparing video presentation;
- Preparing report.

Zhadyraiym Akunova (6% of work)

- Reviewing literate on the topic;
- Dataset choosing and preprocessing;
- Preparing video presentation.

E. Reproducibility checklist

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: 100/100. Code for experiments were written on our own, but we used open-source libraries (e.g. sklearn, lightgbm, miceforest) with all credits paid

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

9. The exact number of evaluation runs is included.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None