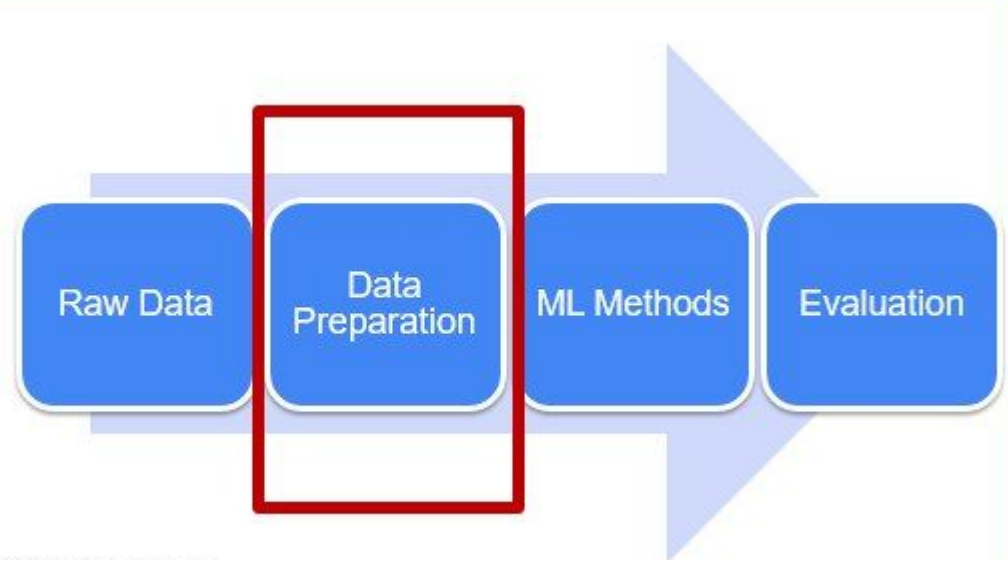
# Missing values and noise in data. How crucial it can be?

Team “Do not miss your value”

Vladislav Molodtsov
Irina Shushpannikova
Stepan Vasilev
Kelvin Kutsukutsa
Zhadyraiym Akunova

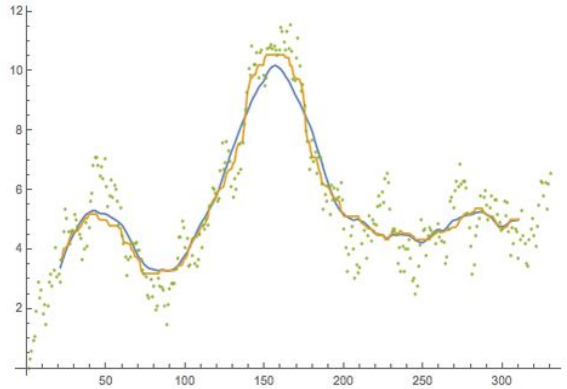
# Introduction



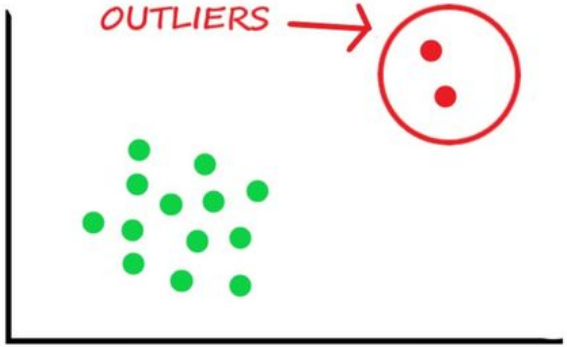

*Data preparation is important*

# Introduction

**Noise**




**Outliers**




**Missing values**

*How crucial is it?*

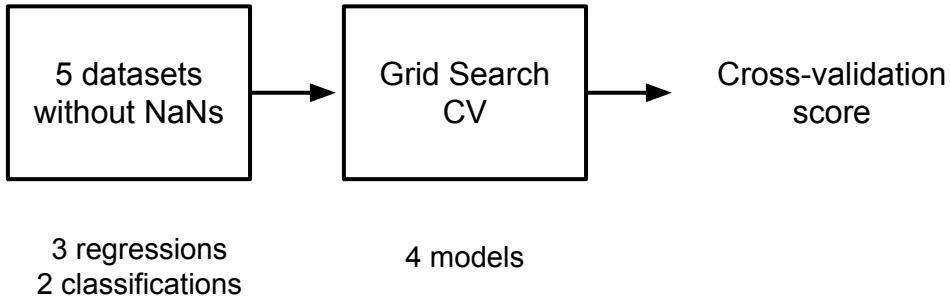
# Goals

- Figure out how noise and missing values affect the quality of ML models
- Compare different imputation techniques
- Find out how different imputation techniques are affected by noise

# Outline

- Datasets, models, metrics
- Experiments with noise only
- Experiments with missing values only
- Experiments with both noise and missing values
- Conclusions and recommendations

# Pipeline of experiments (1/4)




Find hyperparameters for models

Compute score on initial datasets

# 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 |

5 datasets from different domains

Samples with missing values are manually removed

# Hyperparameters tuning

- 4 ML models:
  - Linear Regression or Logistic Regression (Linear)
  - Decision Tree (DT)
  - Random Forest (RF)
  - Gradient Boosting (LightGBM)
- 3-Fold Cross-Validation
- Grid Search parameters:
  - n_estimators = {10, 20, 30, …, 100}
  - max_depth = {3, 5, 7}
  - C = {1e-3, 1e-2, …, 1e3}

# Cross-validation score

Classification

*F1-micro*

F1 takes into account both precision and recall
Micro averaging is preferable for imbalance datasets

Regression

*Mean Absolute Percentage Error (MAPE)*

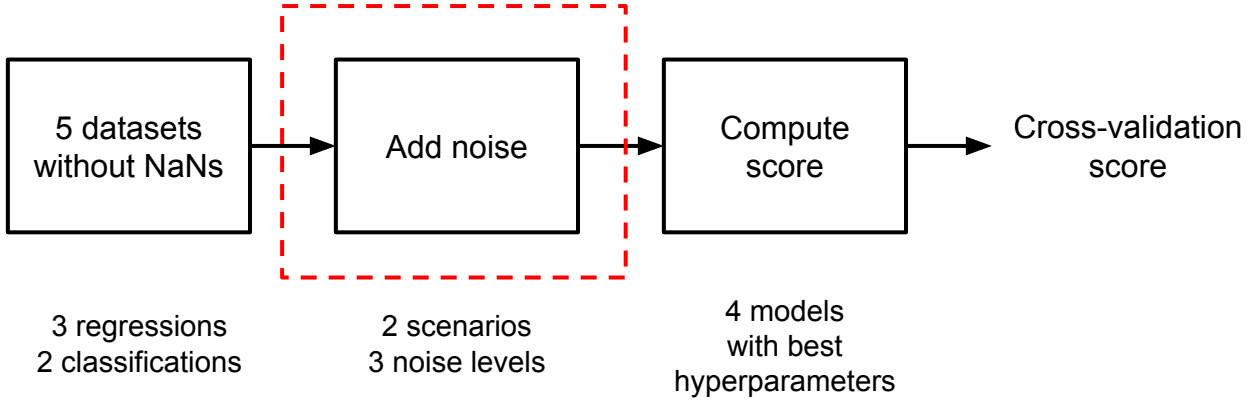
Is not sensitive to scale of target

# Cross-validation scores on initial datasets

| | Dataset name | Problem | Score name | Cross-val 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** |

*Note that score does not change a lot across models in most cases*

# Pipeline of experiments 2/4




How score is affected by noise?

# Adding noise

Additive white Gaussian noise (AWGN) with predefined Signal-to-Noise Ratio (SNR)

- Iterate over all continuous features
- Compute average square of feature (“signal power”)
- Add noise with the power corresponding to the defined SNR

21 levels of SNR: -20dB, …, 20dB

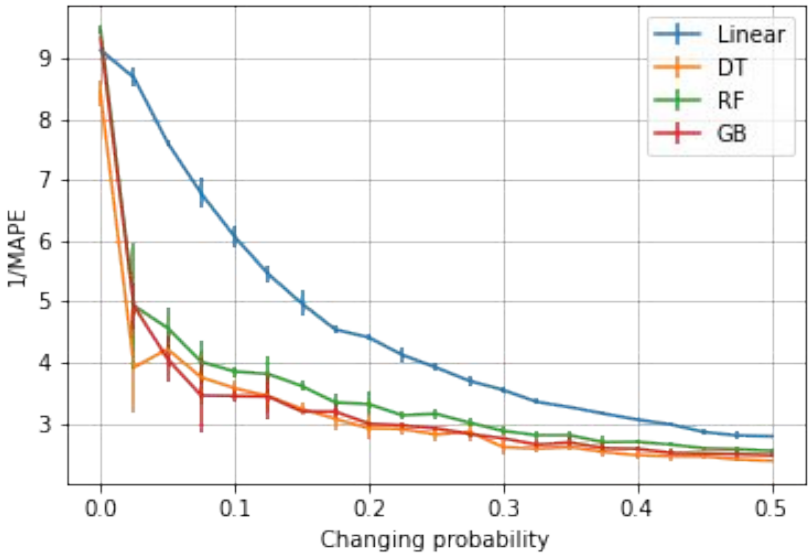
Changing value of each sample with predefined probability *p*

- Iterate over all features
- If feature is categorical, change its value to another one with probability *p*
- If feature is continuous, change its value to some value from min to max with probability *p*

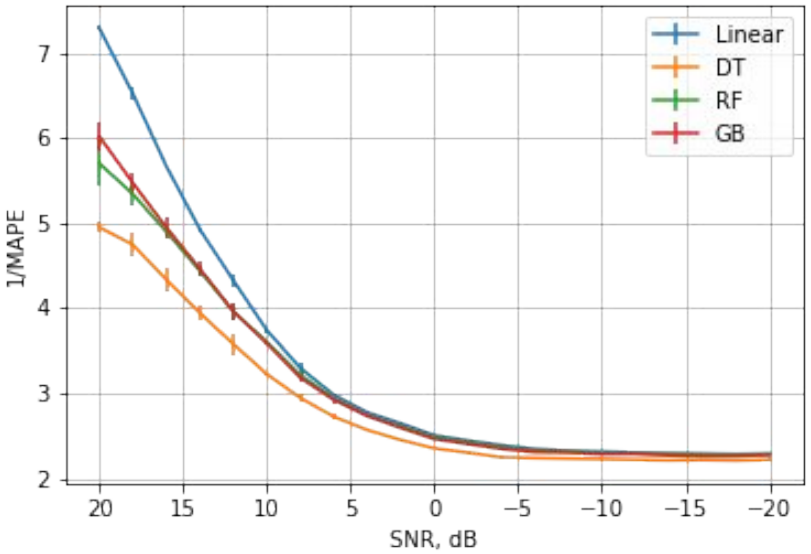
21 values of p: 0, …, 0.5

# Results dataset 3 (regression)

Value changing at random



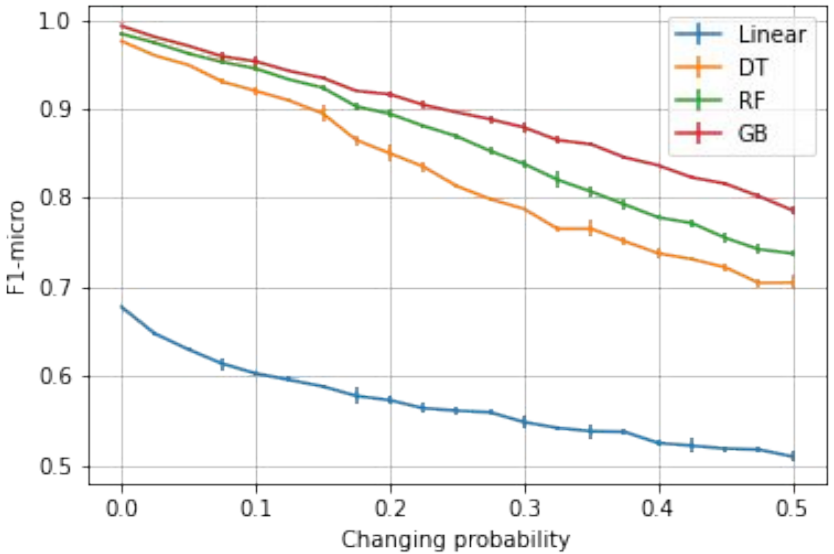

AWGN noise



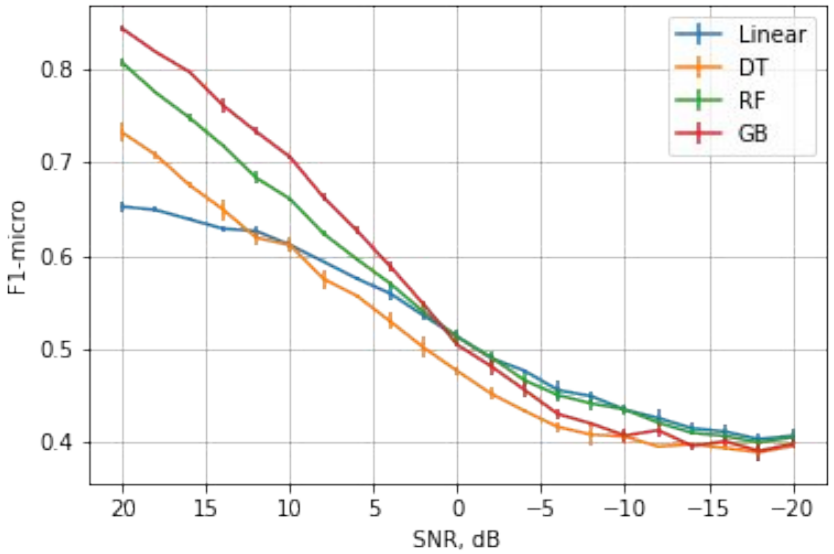

Linear model outperforms other while RF and GB fail

# Results dataset 5 (classification)

Value changing at random




AWGN noise

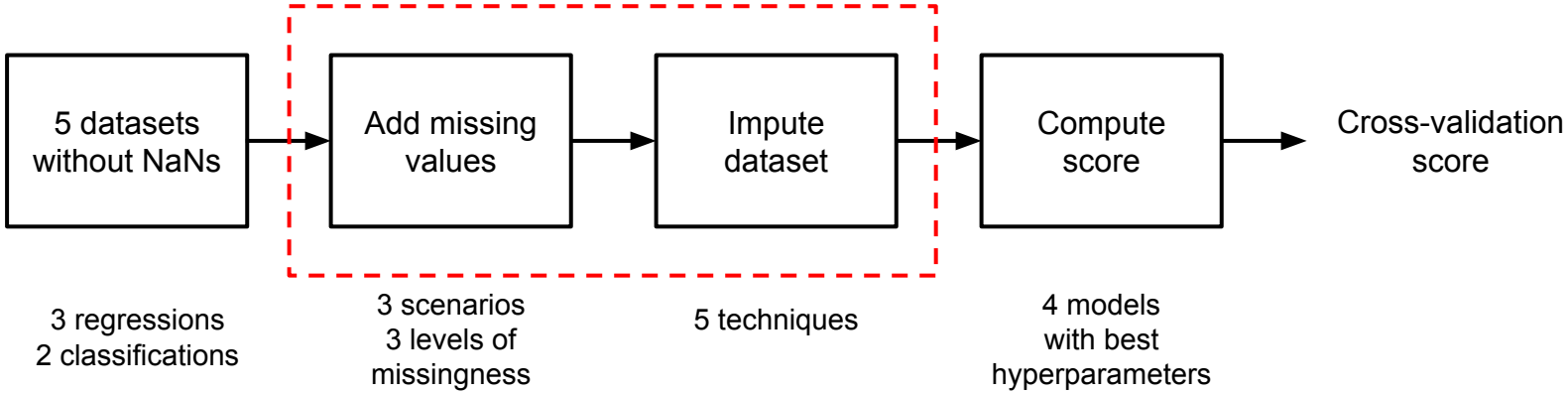



GB outperforms other models while linear model fails

# Findings

- Noise can significantly affect ML models quality
- Models behave differently on different datasets in case of noise
- For some datasets, noise can be more crucial than for others

# Pipeline of experiments 3/4




How score is affected by missing values?

# Adding missing values (1/3)

**Missing Completely at Random (MCAR)**

$$P(R|D^m, D^o) = P(R)$$

- Iterate over all rows and columns
- Drop every entry with probability *p* = {0, …, 0.5}

# Adding missing values (2/3)

**Missing at Random (MAR)**

$$P(R|D^m, D^o) = P(R|D^o)$$

- Randomly split the dataset into two equal parts Ai and *Aj*
- Drop entries *Aj* depending on the corresponding entry in *Ai* with probability 4p
- Numerical features are compared with median value
- Categorical features are compared with selected classes

# Adding missing values (3/3)

**Missing Not at Random (MNAR)**

$$P(R|D^m, D^o) = P(R|D^m)$$

- Iterate over all features
- Numerical values are dropped if they less (larger) the median value
- Categorical features are dropped if they belong to randomly selected classes

# Imputation techniques

- Filling with 0
- Filling with mean
- Filling with median
- Filling by MICE (Multiple Imputation by Chained Equation)
  - Iteratively predict missed features by others using Random Forest
- Filling by KNN (K-Nearest Neighbor imputation)
  - Predict missed features by mean value of 5 nearest neighbors

# Distortion measurement

Classification

$$\frac{F1_{distorted}}{F1_{initial}}$$

Regression

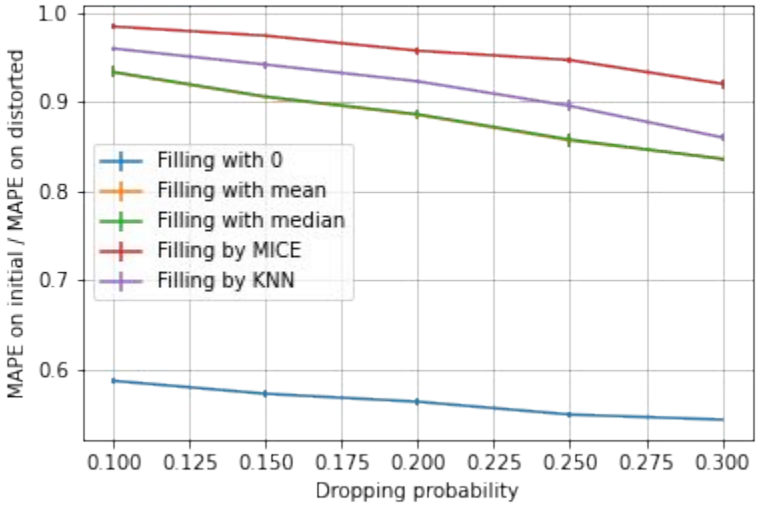
$$\frac{MAPE_{initial}}{MAPE_{distorted}}$$

*Meaning: how the dataset quality decreased*

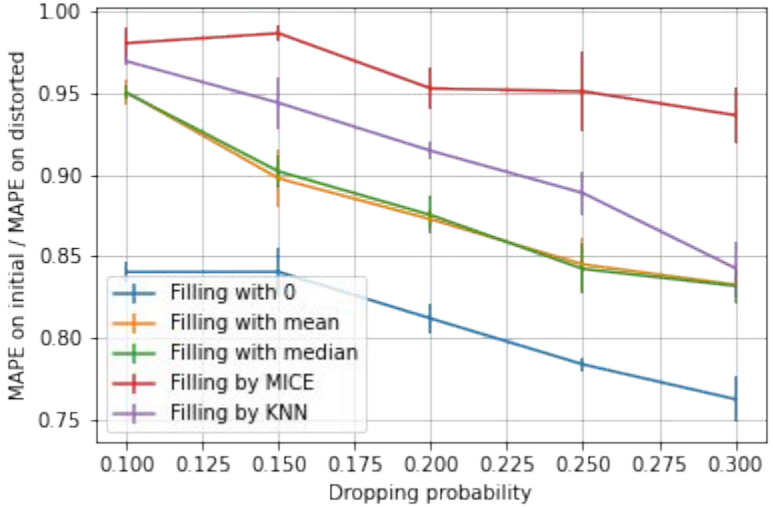
**Good imputation method increases this ratio**

# Results dataset 1 (regression), MAR

Linear model



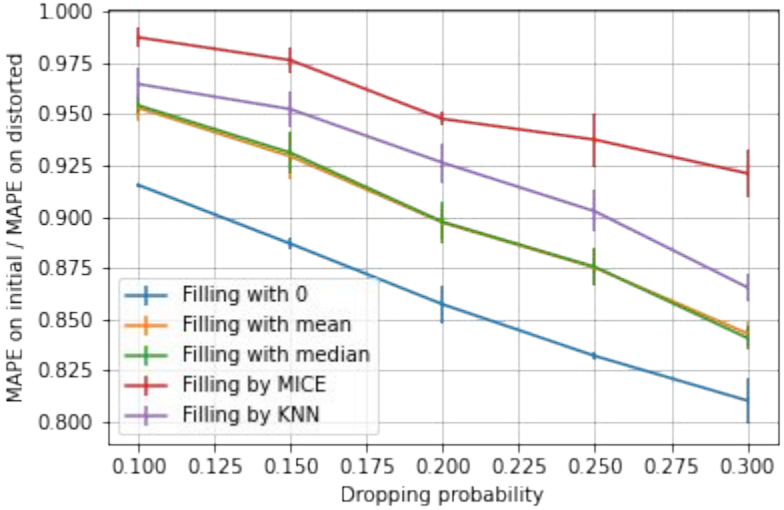

Decision Tree



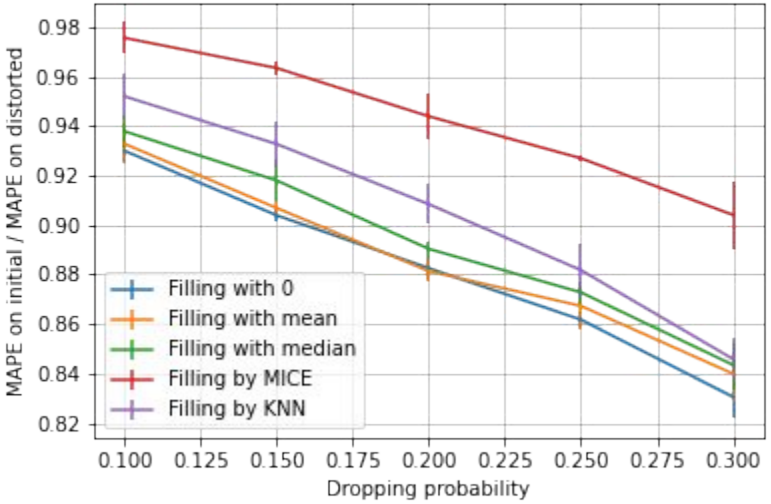

MICE outperforms other imputation methods

# Results dataset 1 (regression), MAR

Random Forest



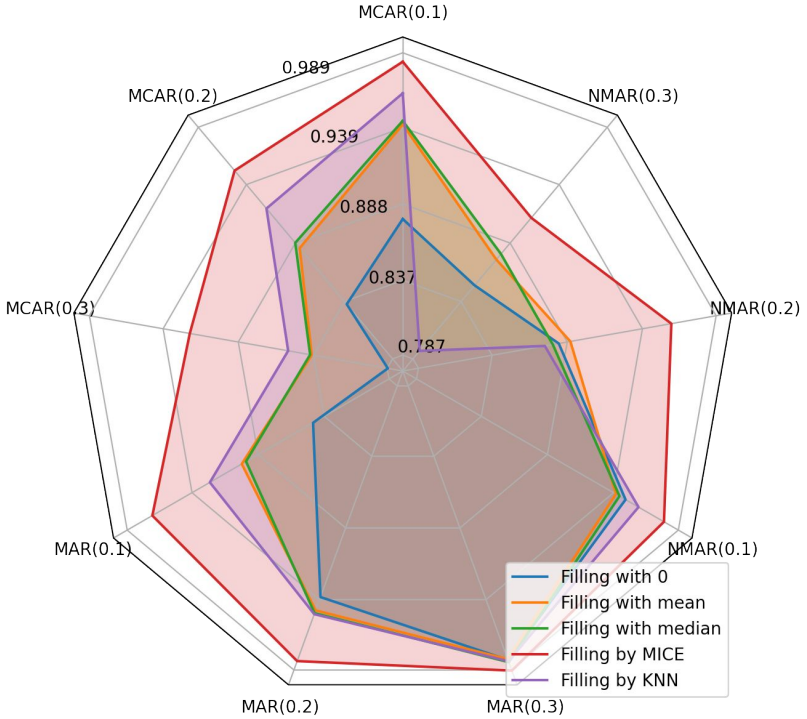

Gradient boosting




MICE outperforms other imputation methods

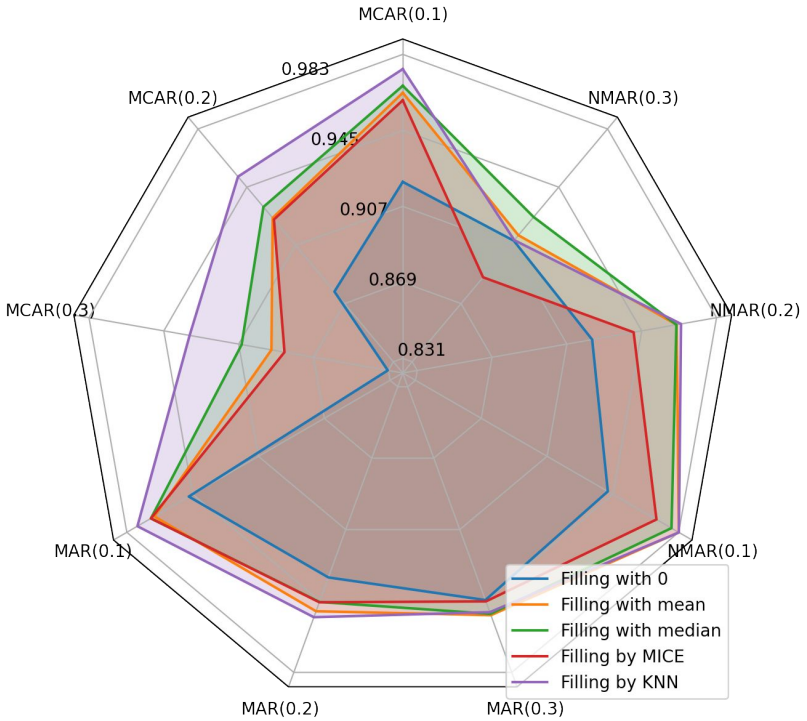
We can average over models!

# Results dataset 1 (regression)




MICE outperforms other imputation methods

# Results dataset 5 (classification)

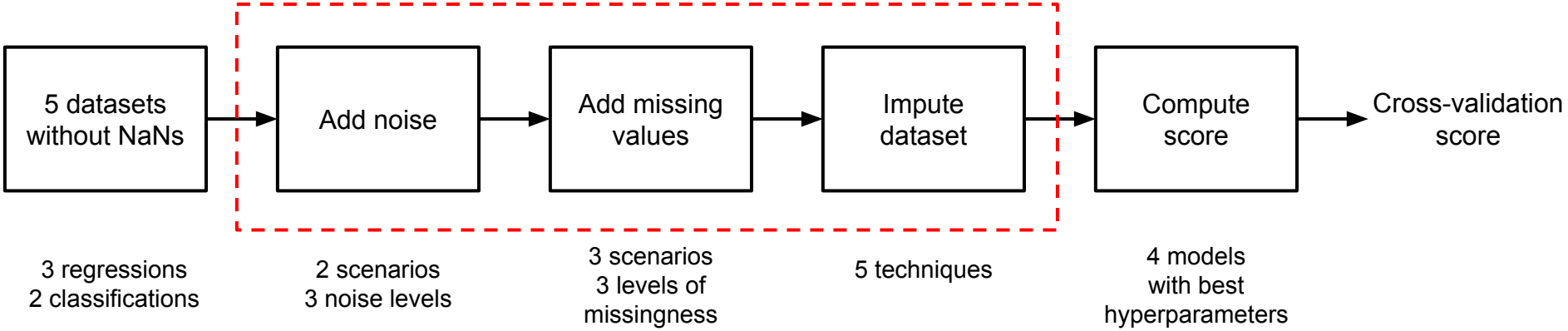



MICE fails while KNN and filling by median are the best

# Findings

- On some datasets, MICE and KNN can perform significantly better than simple imputation methods (0, mean, median)
- On other datasets, MICE and KNN can fail

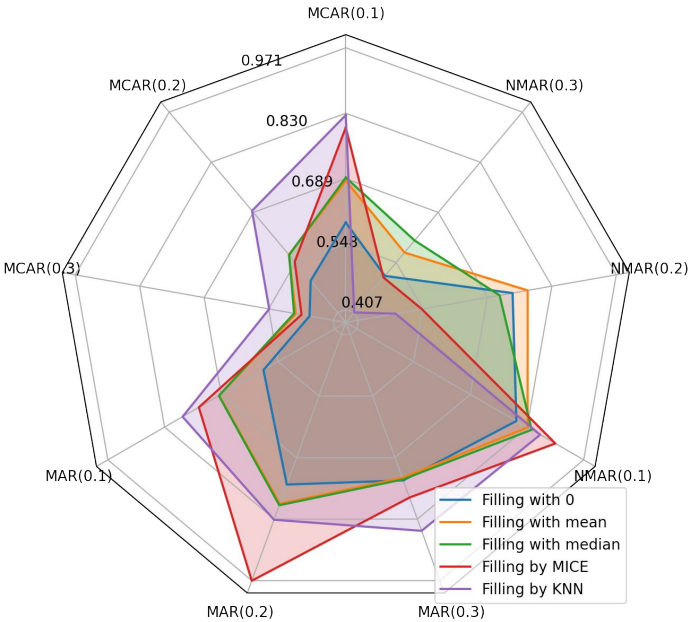
# Pipeline of experiments 4/4



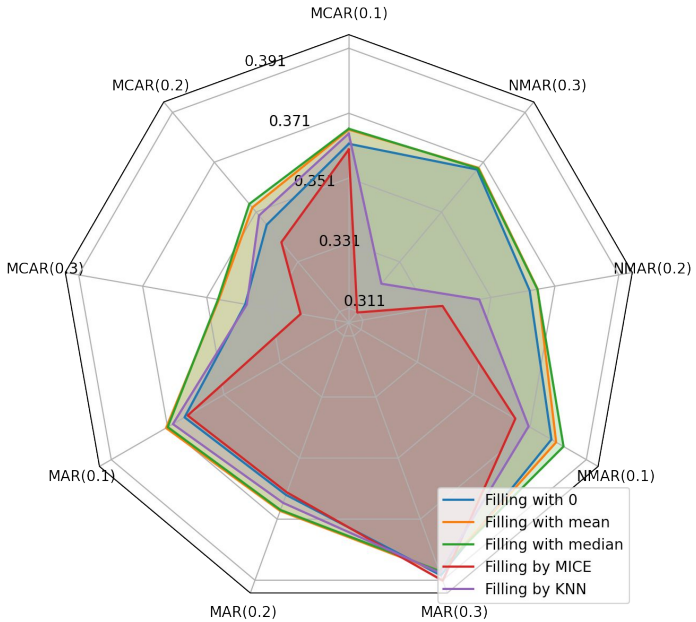

Which imputation method is the best in presence of noise?

# Results dataset 3 (regression)

Without noise



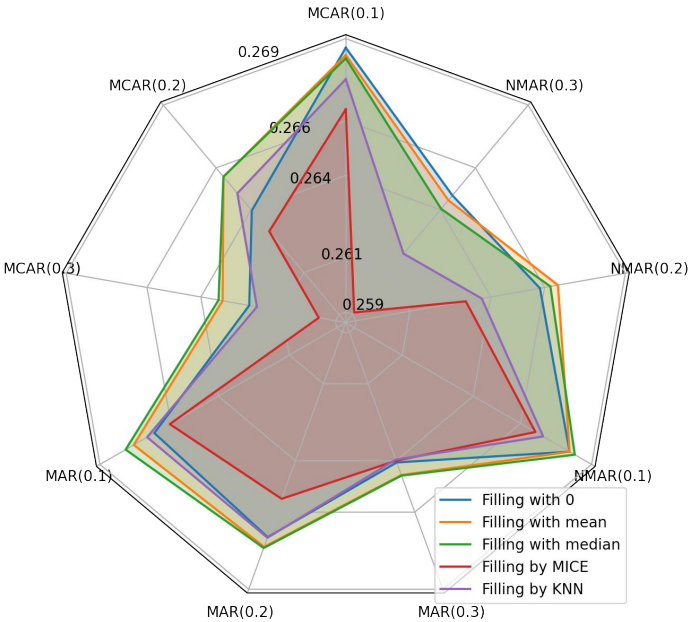

AWGN noise 10dB SNR



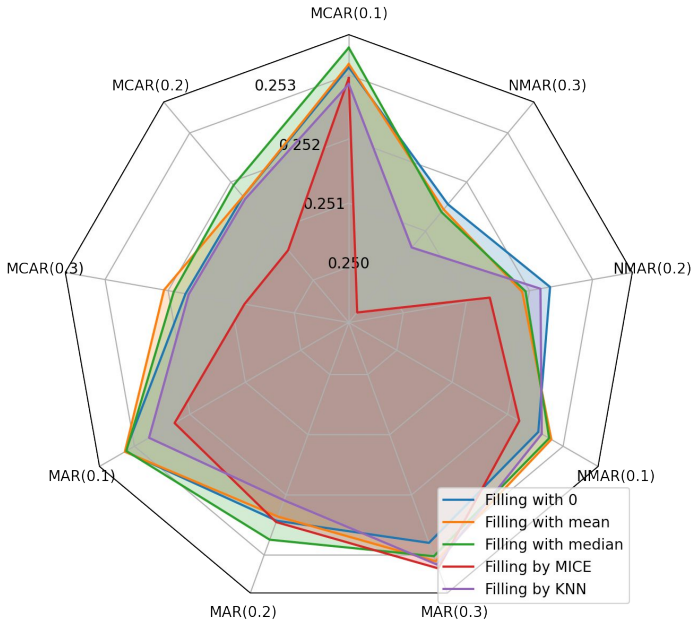

Imputation quality is significantly affected by noise

# Results dataset 3 (regression)

AWGN noise 0dB SNR




AWGN noise -6dB SNR




Behavior of imputation methods is unpredictable

# Findings

- Noise can significantly affect imputation quality
- It is unpredictable which imputation method is the best in presence of noise

# Conclusions

- Noise and missing values can significantly affect the quality of ML models
- In some cases, sophisticated imputation techniques, such as MICE and KNN, can give huge gain
- In other cases, MICE and KNN fail
- Noise can significantly affect imputation quality
- It is unpredictable which imputation method is the best for particular level of noise

# Recommendation

- Eliminate noise and missing values as much as possible
- Try more sophisticated imputation techniques such as MICE and KNN instead of just using filling by 0 or mean
- Cross-validate among different imputation options

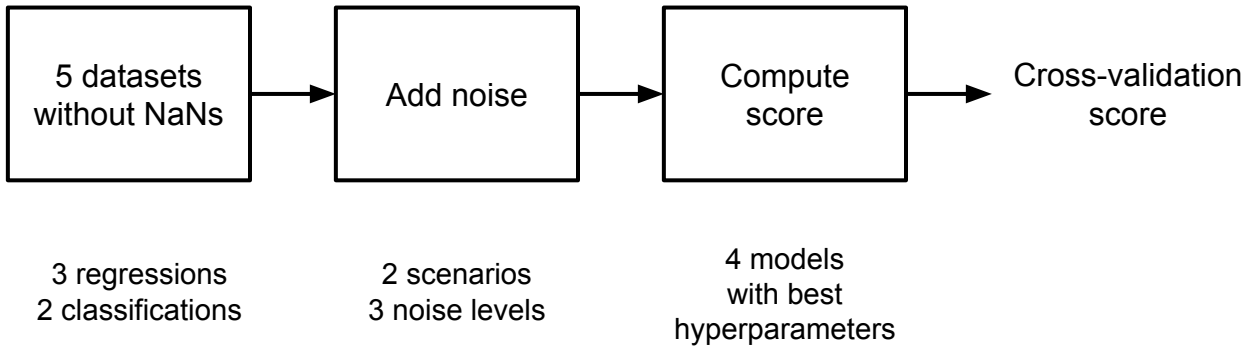
Thank you for your attention!

# Backup slides

# Experiments configuration

- 2 problems: regression & classification
- 5 datasets in total
- 2 scenarios of adding noise:
  - AWGN noise with 41 levels of SNR: -20dB, …, 20dB
  - Random changing with 21 probabilities: 0, …, 0.5
- 3 scenarios of adding missing values: MCAR, MAR, NMAR
- 3 probabilities to miss value: 0.1, 0.2, 0.3
- 5 imputation techniques: 0, mean, median, MICE, KNN
- 4 ML models: Linear, DT, RF, LightGBM

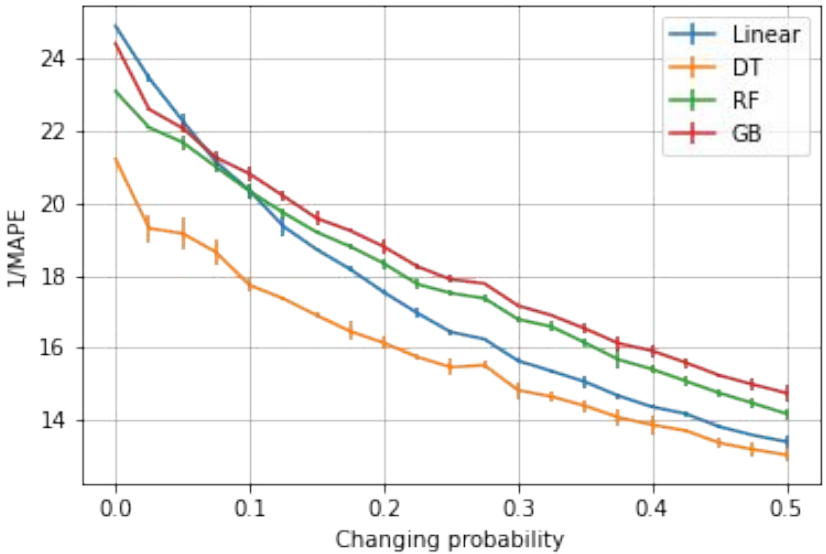
# Pipeline of experiments 2/4



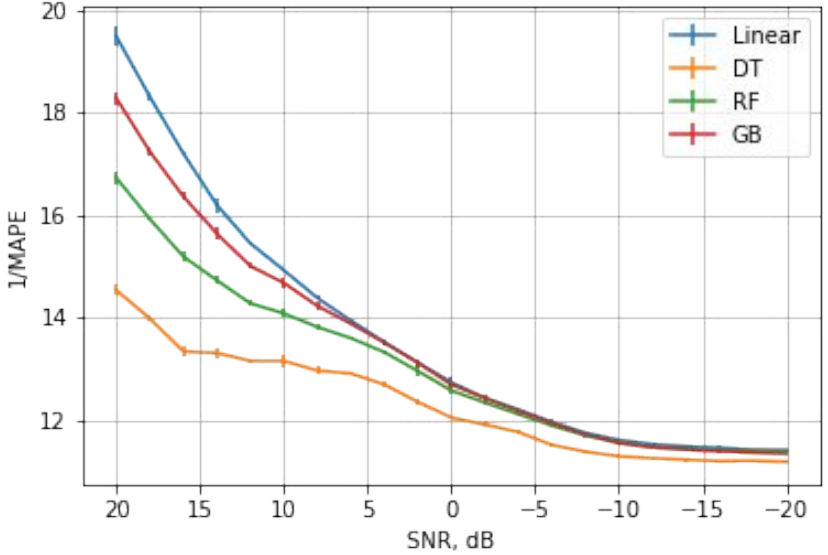

How score is affected by noise?

# Results dataset 1 (regression)

Value changing at random

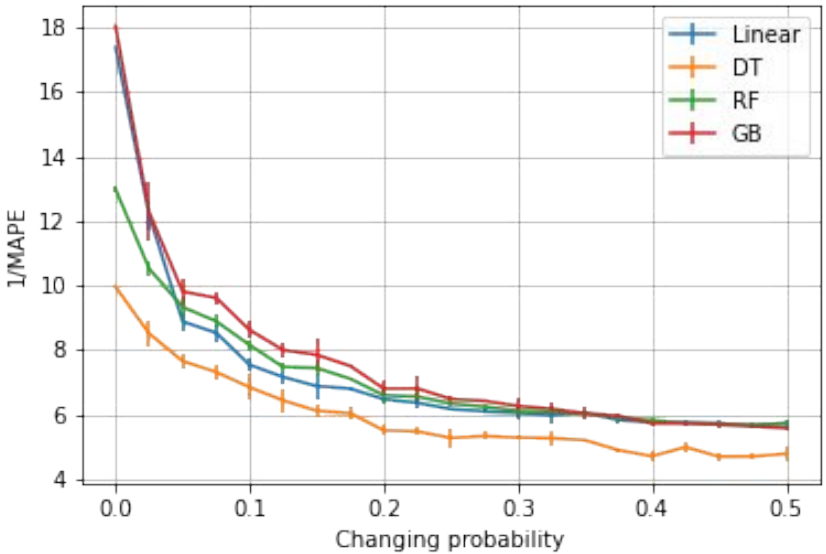



AWGN noise

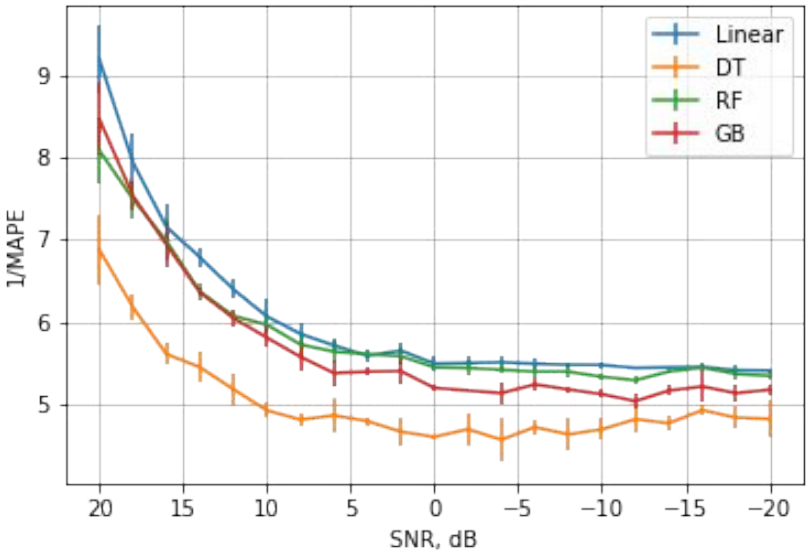


# Results dataset 2 (regression)

Value changing at random

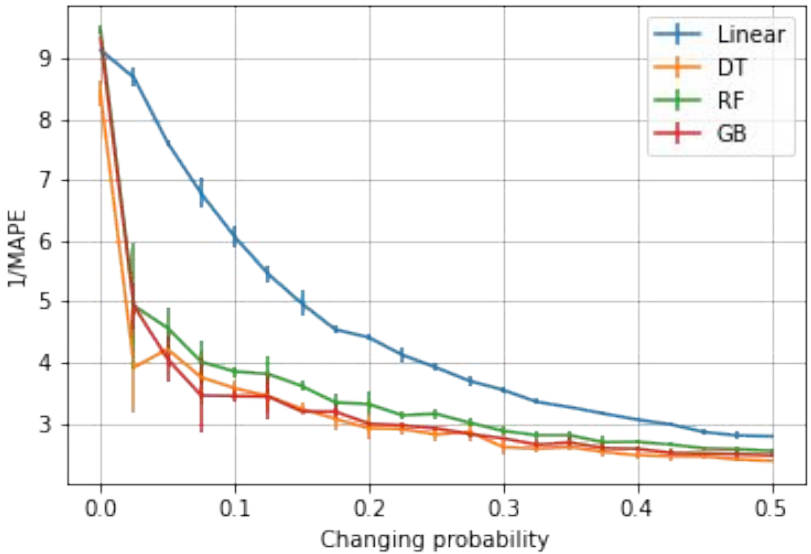



AWGN noise

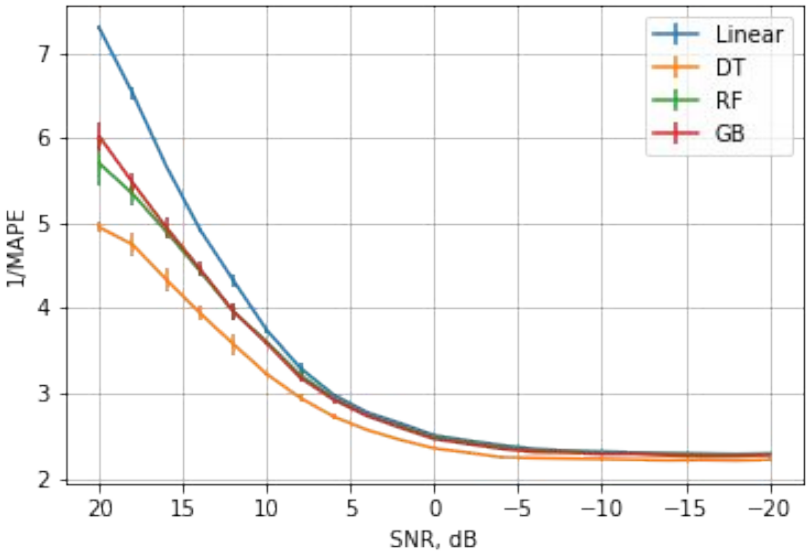


# Results dataset 3 (regression)

Value changing at random

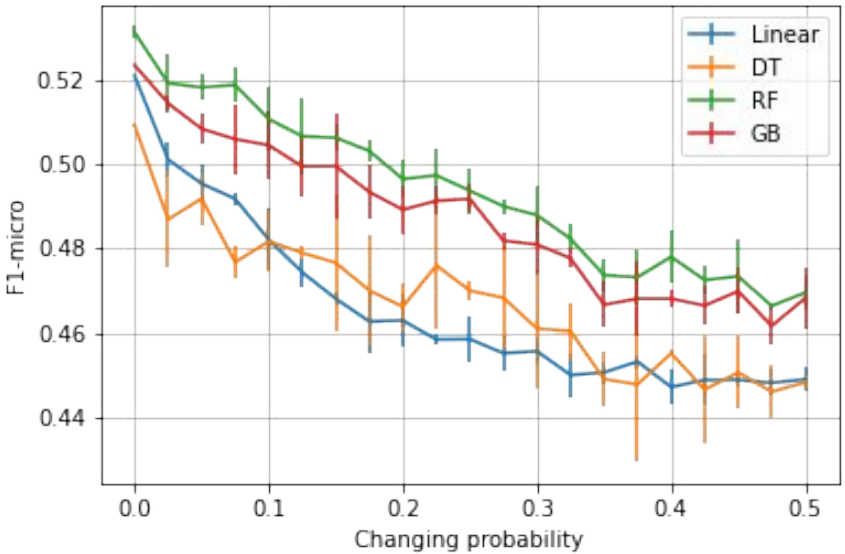



AWGN noise

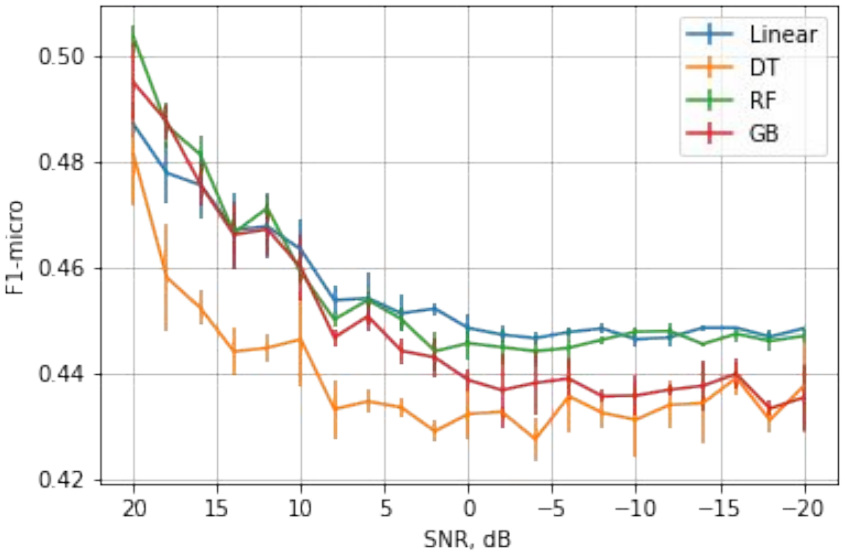


# Results dataset 4 (classification)

Value changing at random

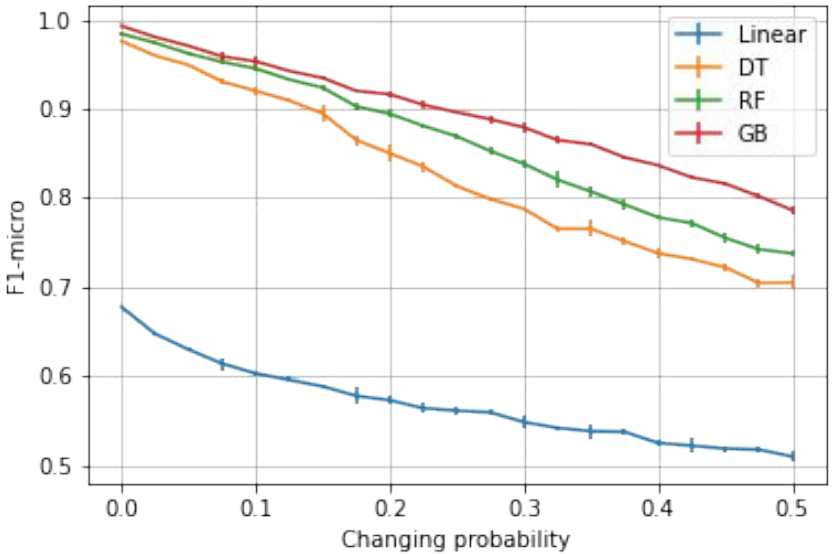



AWGN noise

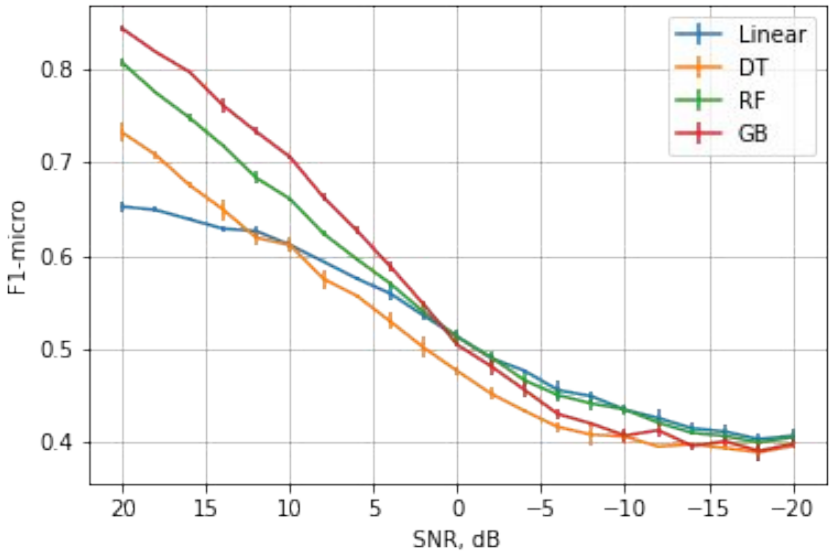


# Results dataset 5 (classification)

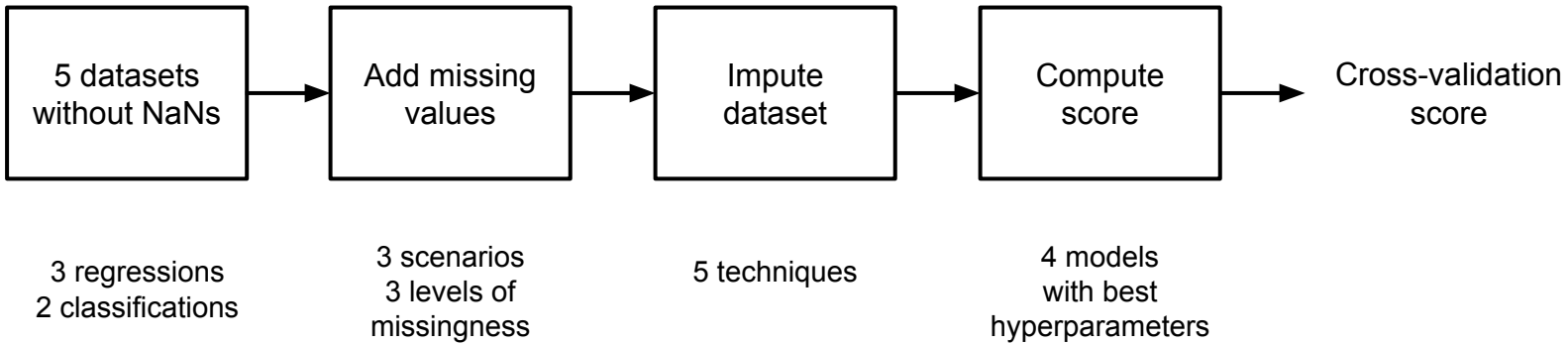
Value changing at random




AWGN noise



# Pipeline of experiments 3/4




How score is affected by missing values?

# Distortion measurement

Classification

$$\frac{F1_{distorted}}{F1_{initial}}$$

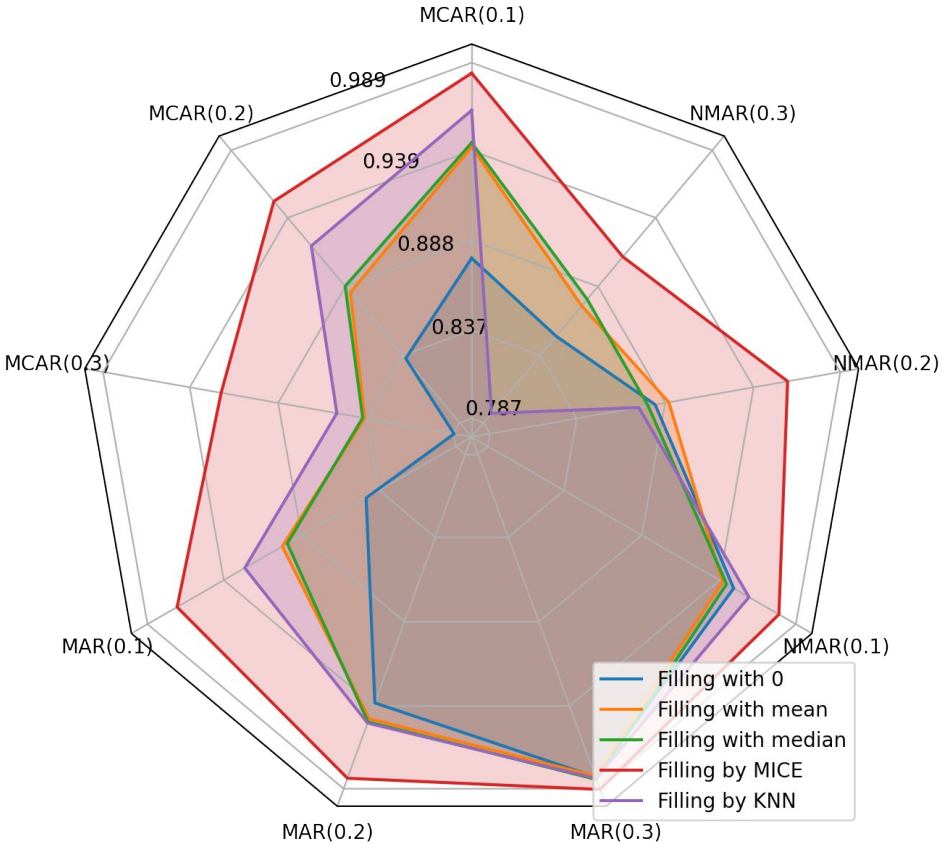
Regression

$$\frac{MAPE_{initial}}{MAPE_{distorted}}$$

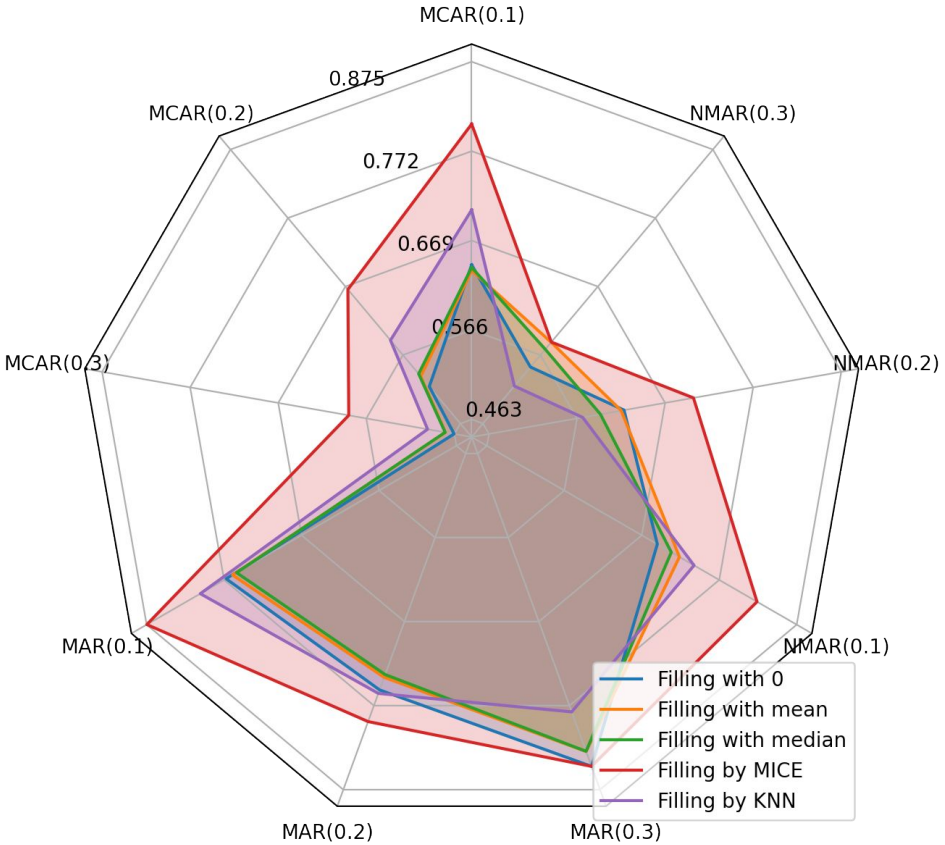
*Meaning: how the dataset quality decreased*

**Good imputation method increases this ratio**

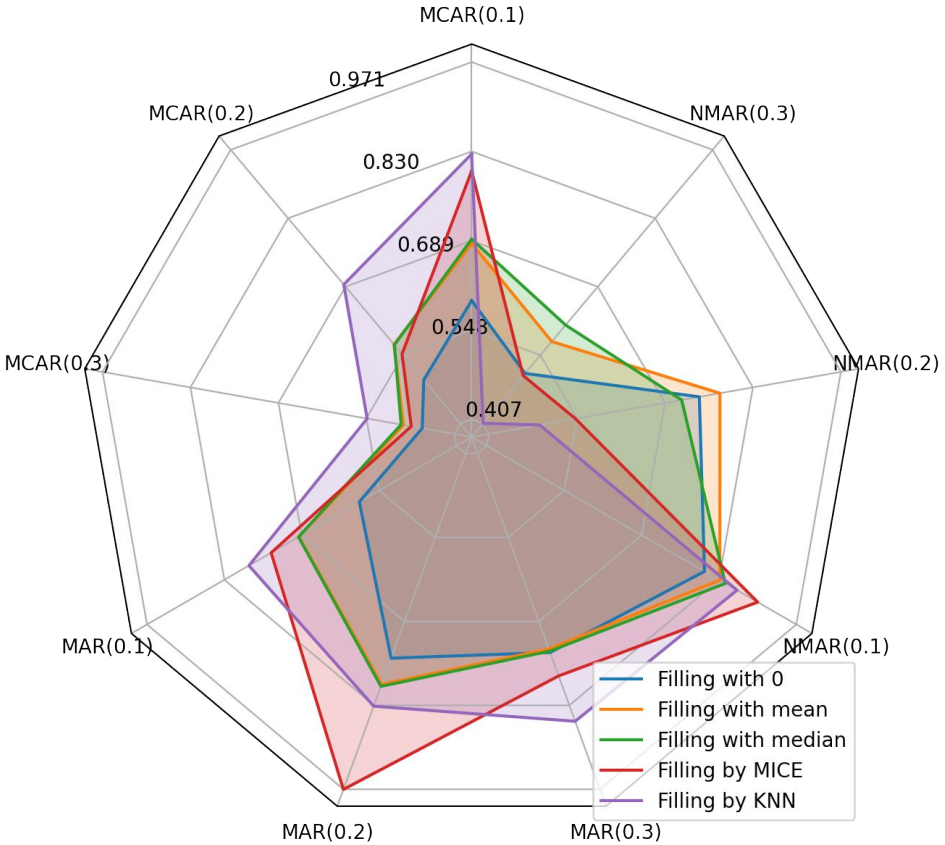
# Results dataset 1 (regression)



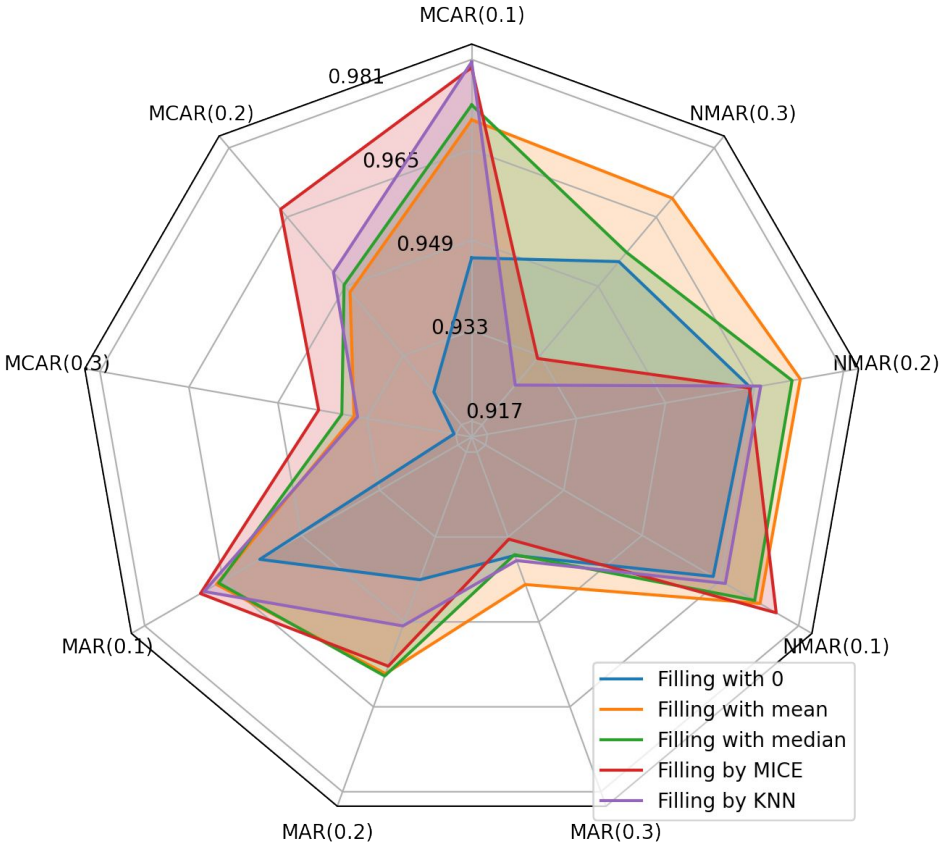
# Results dataset 2 (regression)



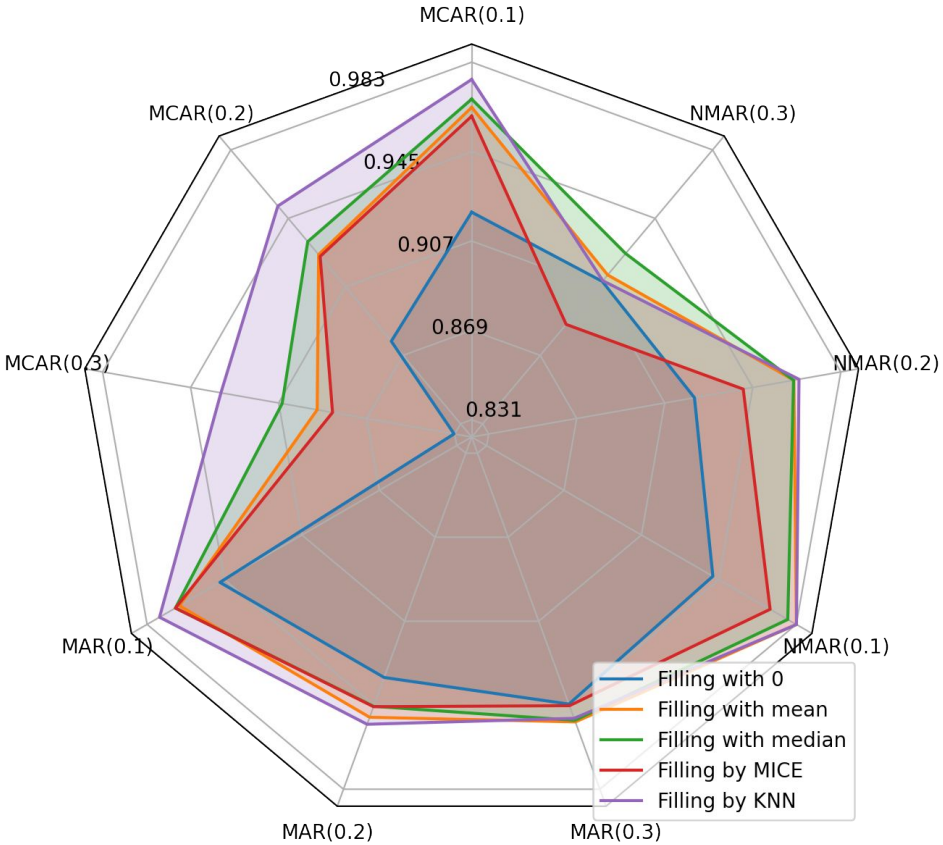
# Results dataset 3 (regression)



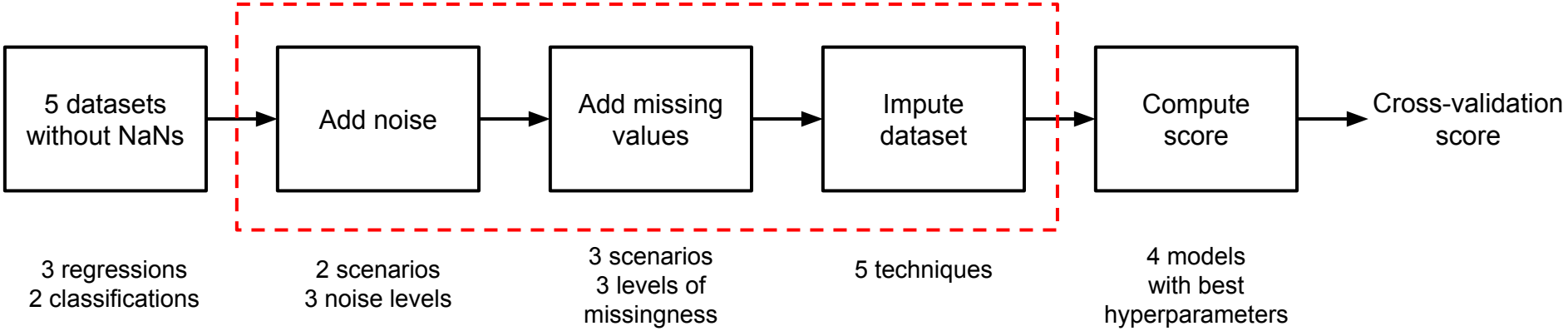
# Results dataset 4 (classification)



# Results dataset 5 (classification)



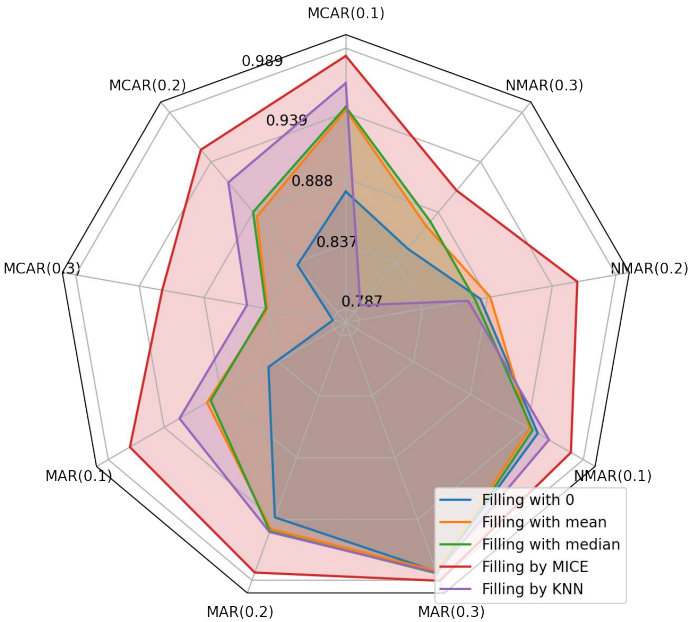
# Pipeline of experiments 4/4



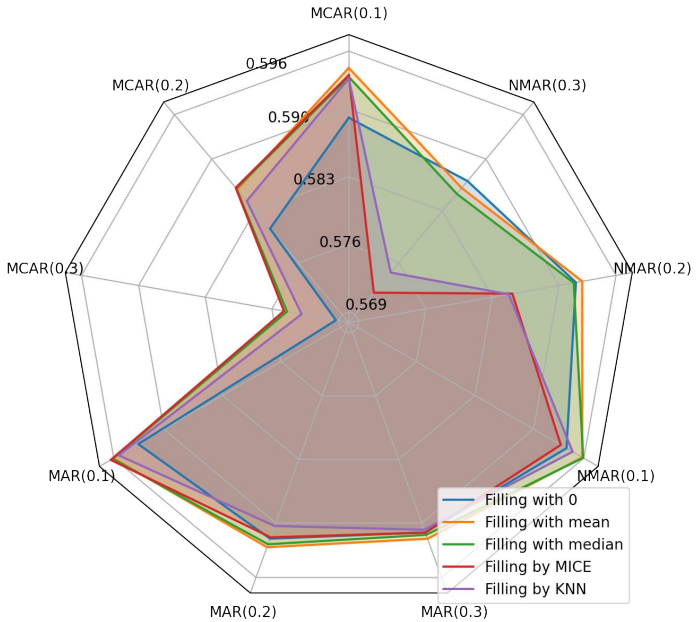

Which imputation method is the best in presence of noise?

# Results dataset 1 (regression)

## Without noise



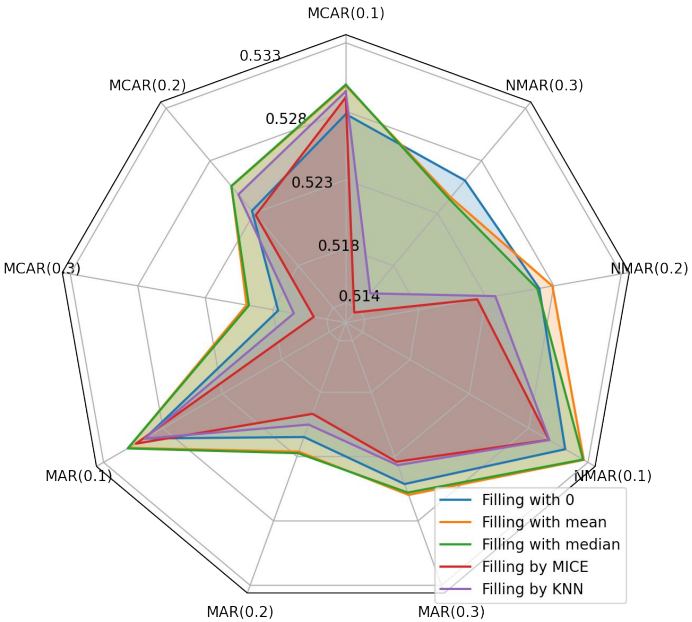

## AWGN noise 10dB SNR



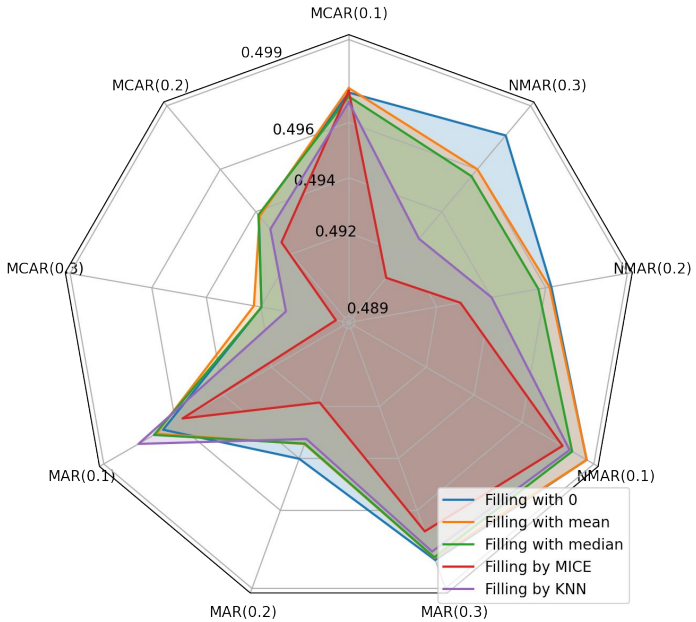

MICE outperforms other imputation methods

# Results dataset 1 (regression)

AWGN noise 0dB SNR



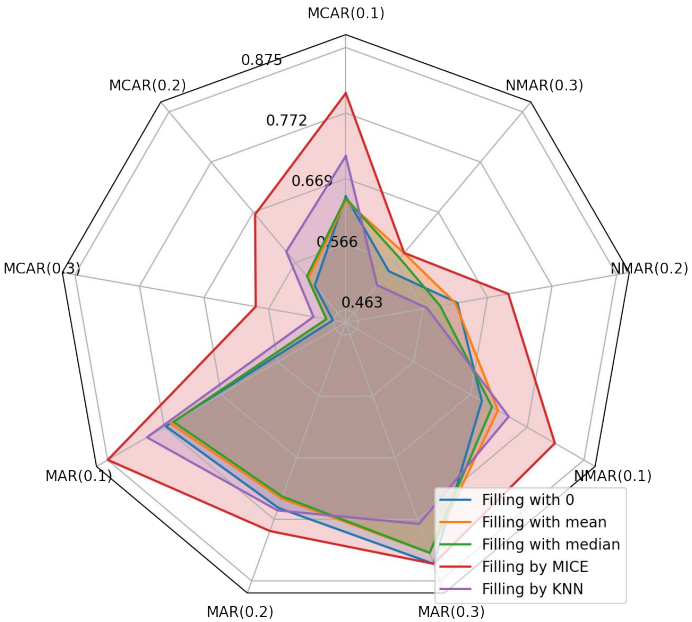

AWGN noise -6dB SNR



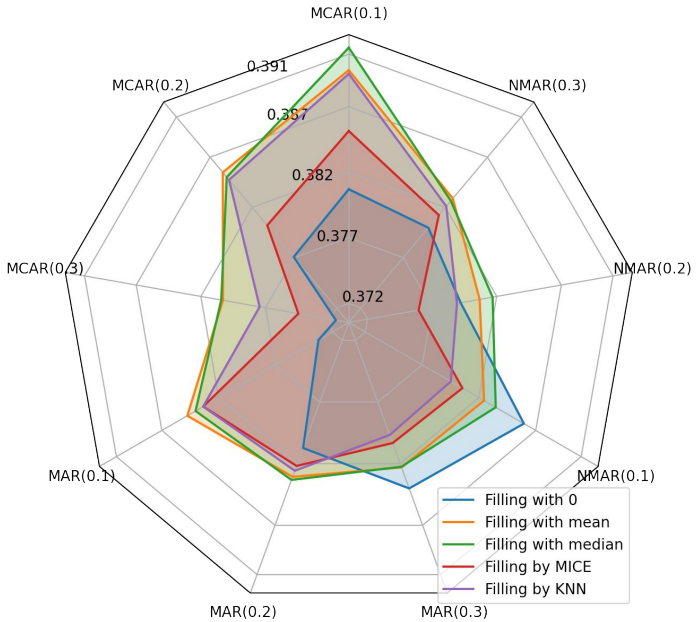

MICE outperforms other imputation methods

# Results dataset 2 (regression)

Without noise



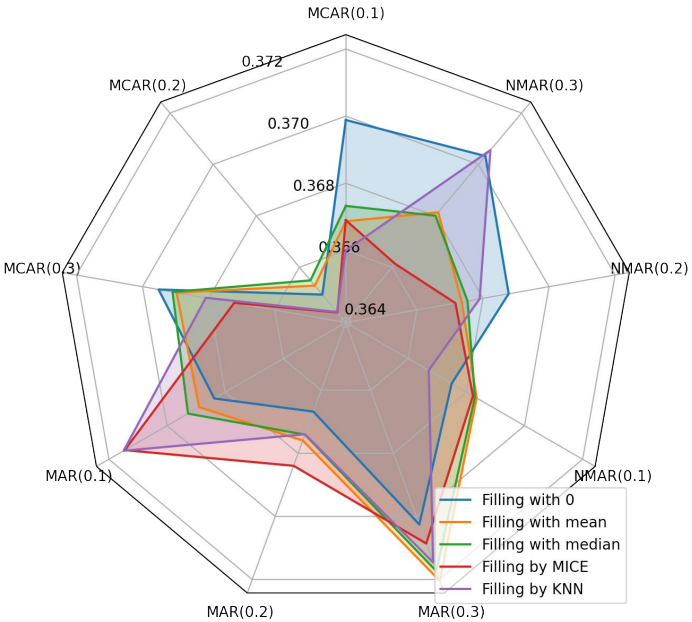

AWGN noise 10dB SNR



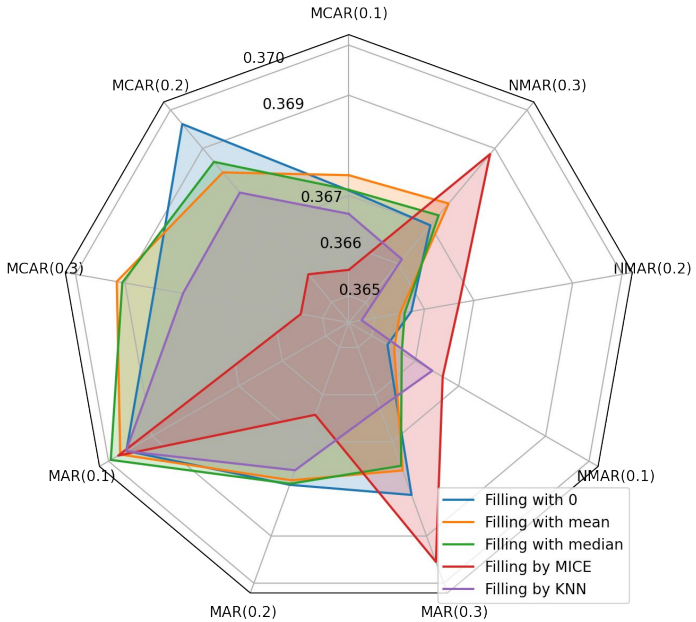

MICE outperforms other imputation methods

# Results dataset 2 (regression)

AWGN noise 0dB SNR



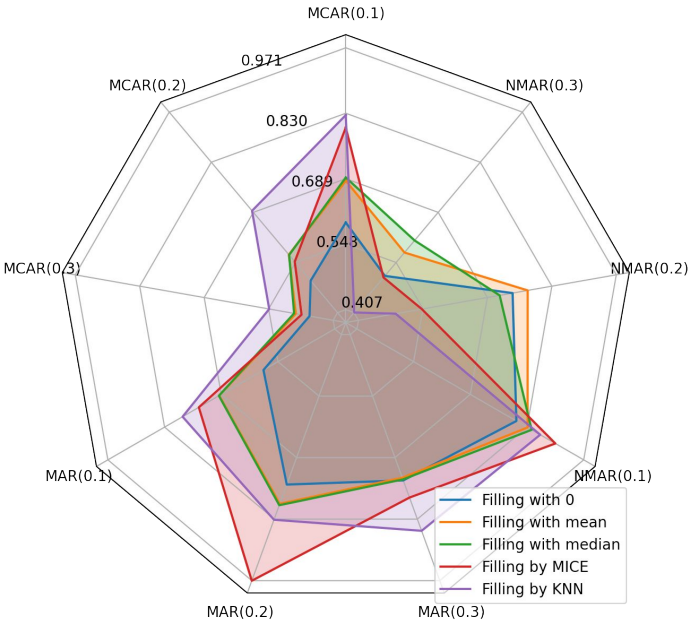

AWGN noise -6dB SNR



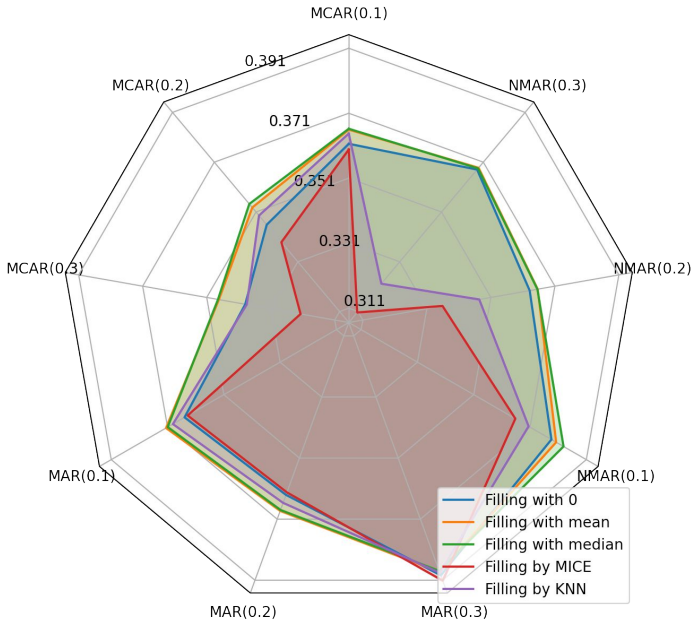

MICE outperforms other imputation methods

# Results dataset 3 (regression)

Without noise



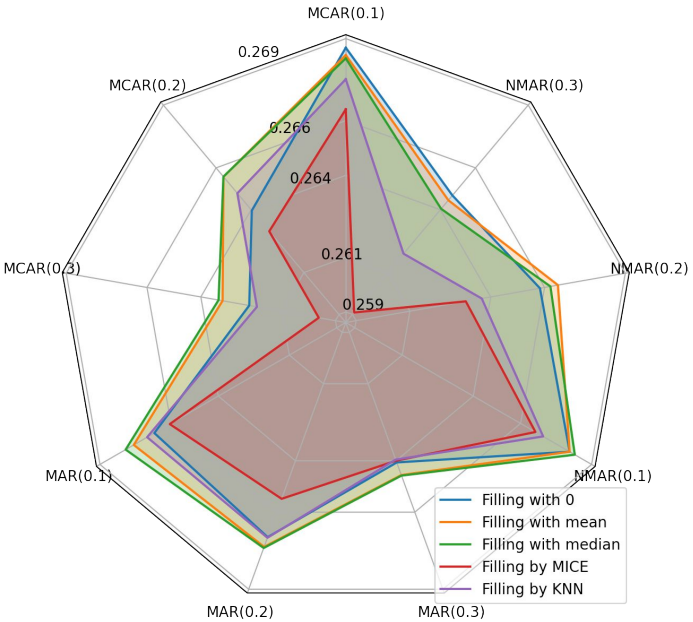

AWGN noise 10dB SNR



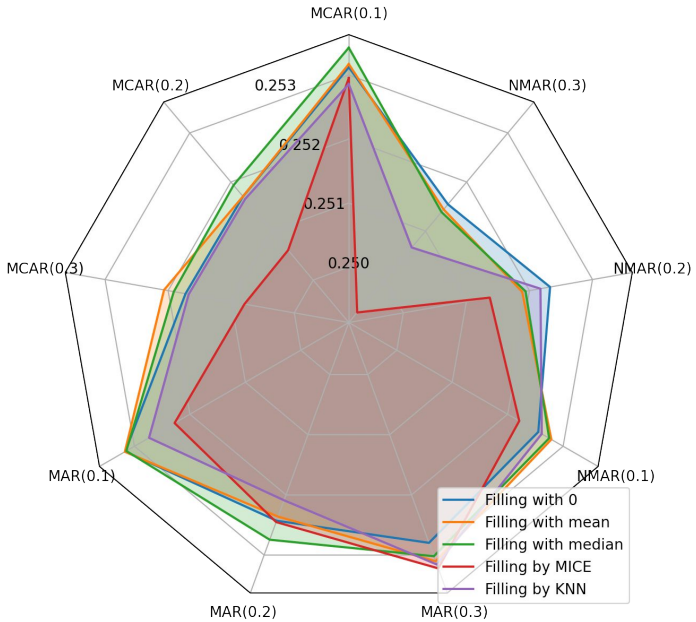

MICE outperforms other imputation methods

# Results dataset 3 (regression)

AWGN noise 0dB SNR



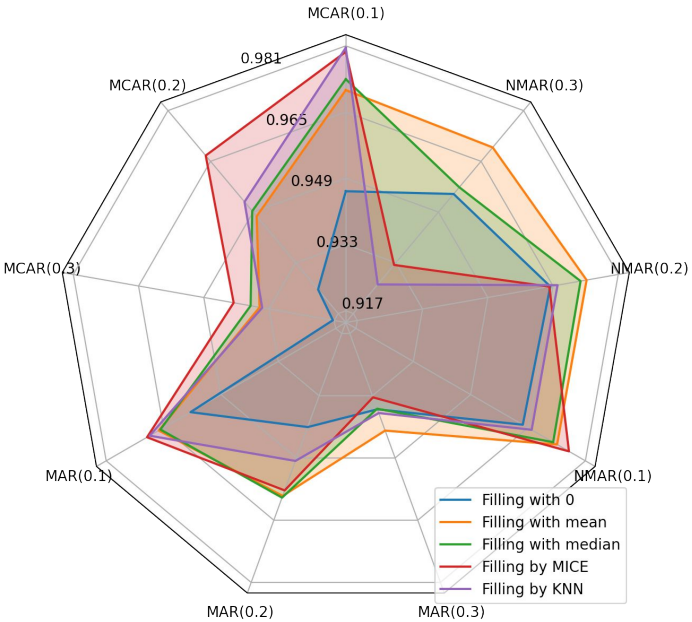

AWGN noise -6dB SNR



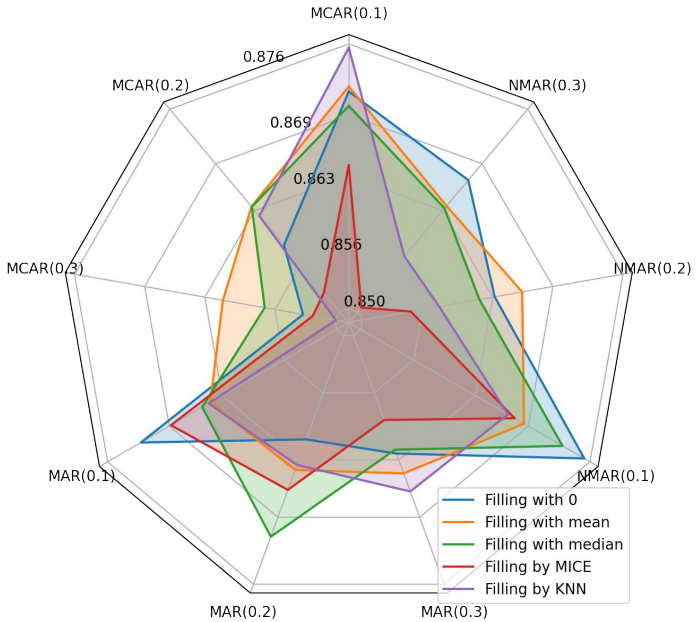

MICE outperforms other imputation methods

# Results dataset 4 (classification)

## Without noise

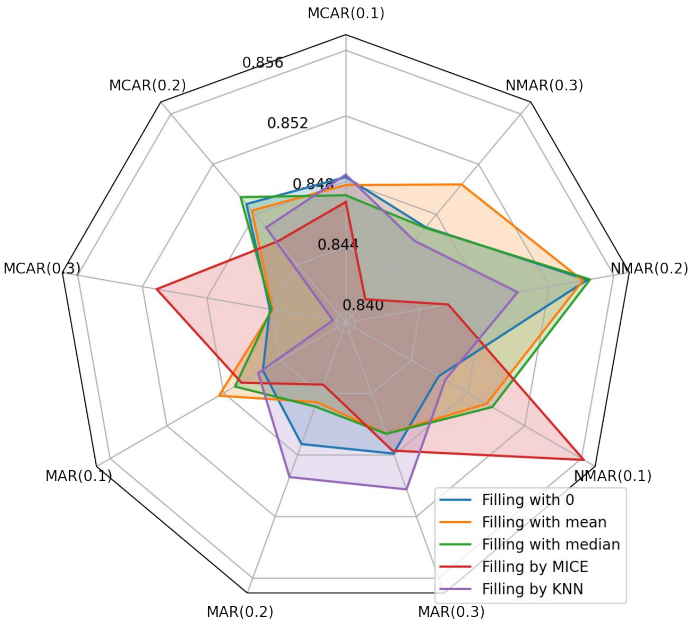



## AWGN noise 10dB SNR

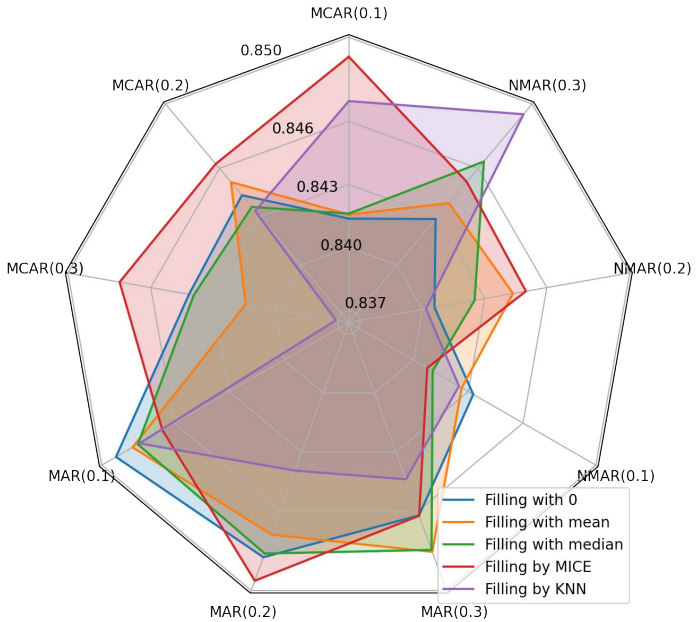


# Results dataset 4 (classification)

AWGN noise 0dB SNR

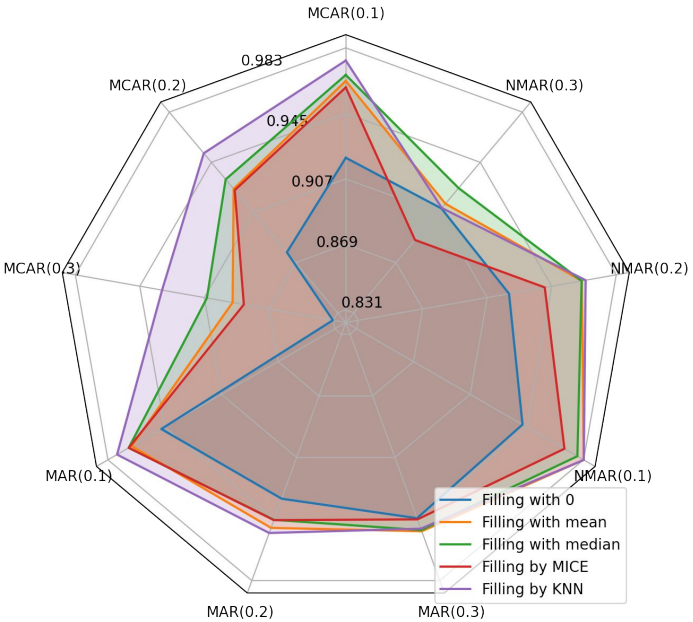



AWGN noise -6dB SNR

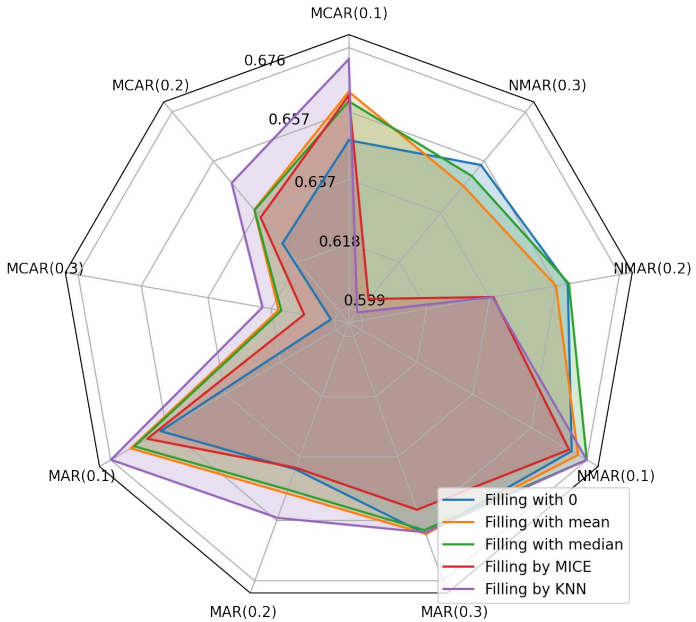


# Results dataset 5 (classification)

Without noise



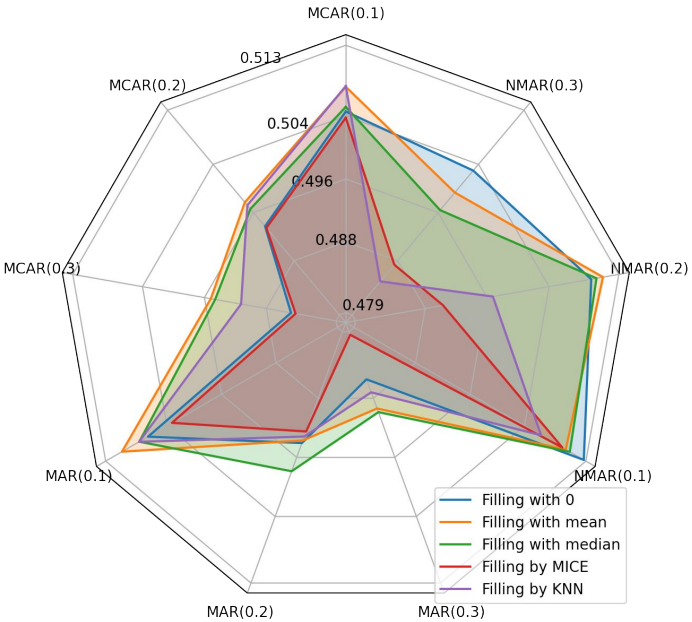

AWGN noise 10dB SNR



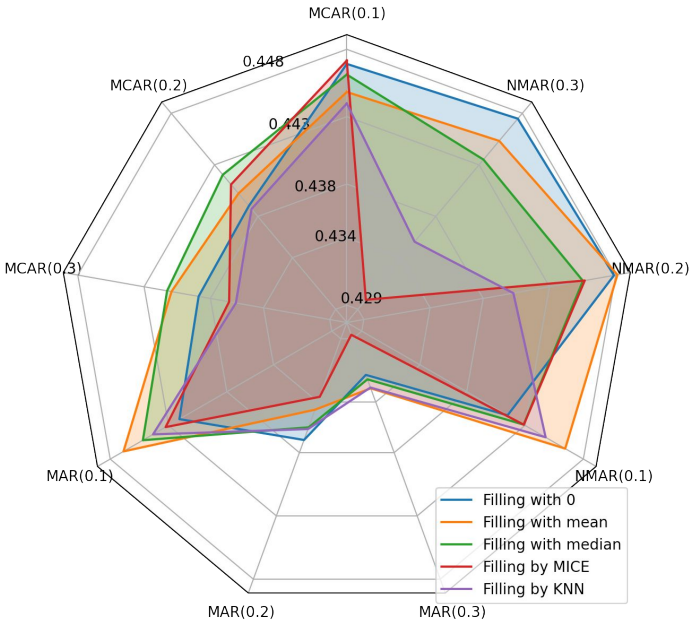

MICE outperforms other imputation methods

# Results dataset 5 (classification)

AWGN noise 0dB SNR




AWGN noise -6dB SNR




MICE outperforms other imputation methods