Creator **Olusanmi Hundogan** Created **Jan 09, 2024, 15:34** Last updated **Jan 09, 2024, 19:23**

Files

- notebooks/analyze_missing_value_imputers_lightgbm.py
- experiments/experiment_missing_value_imputers_lightgbm.py
-

Motivation

This experiment acts as a preliminary study for the experiment which compares different imputation methods. As the LGBMmodel requires considerable time to train we will only use one contender in the overall imputation method comparison. Hence, this experiment acts as preliminary study to choose the best hyperparameters for this task.

Design

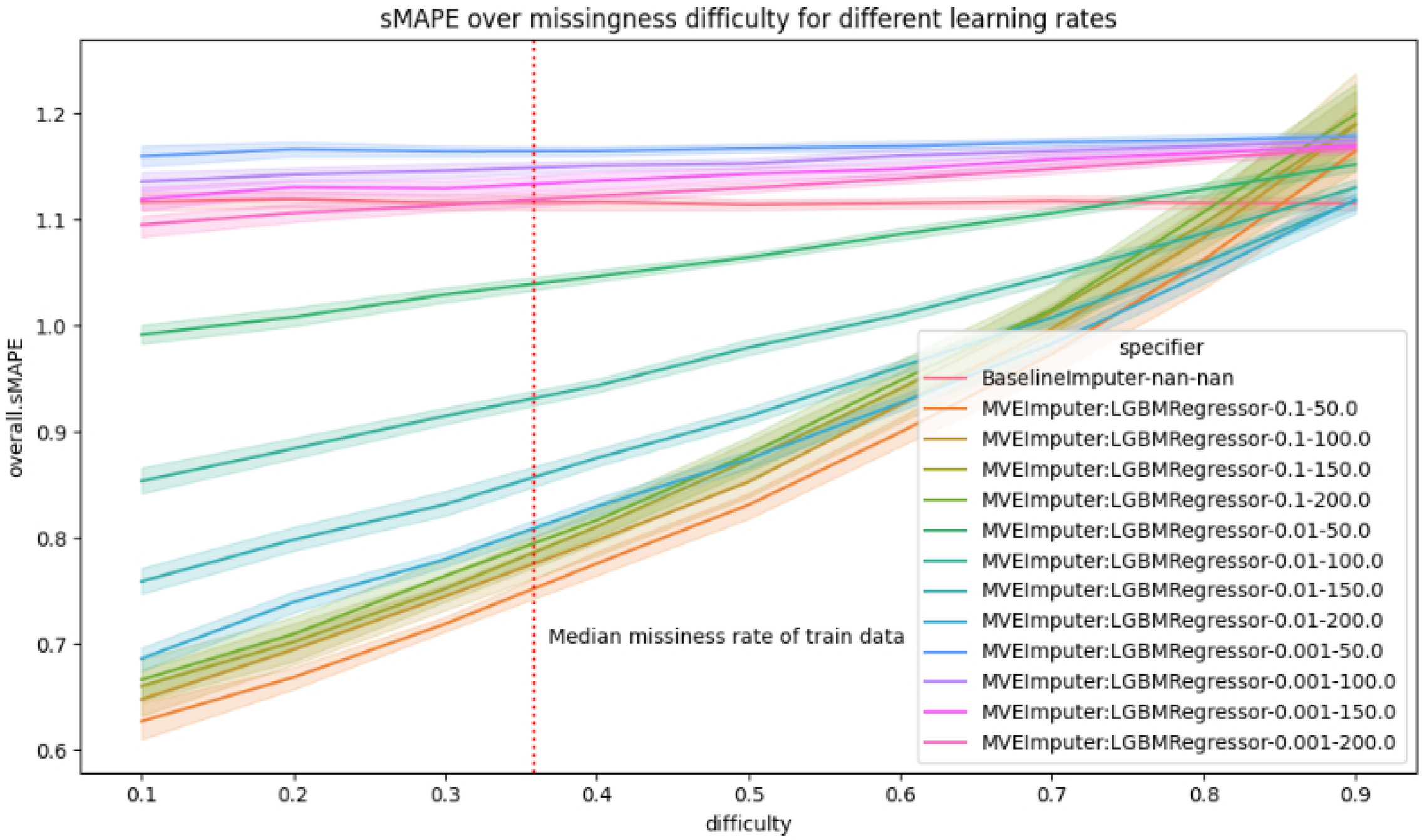
Within the experiment we explored different learning rates and n_estimators. The learning rate values were [0.1, 0.01, 0.001]. The n_estimators parameter space comprises [50, 100, 150, 200] .

We evaluate the imputer by removing values from a dataframe gradually and measuring the sMAPEof reconstructing the values. We test difficulties from 0.1 to 0.9 in even steps. A difficulty of 0.1 corresponds to the artificial removal of 10% of known values. (The number is not completely accurate because the 10% applies to the dataframe, but the dataframe already contains missing values.)

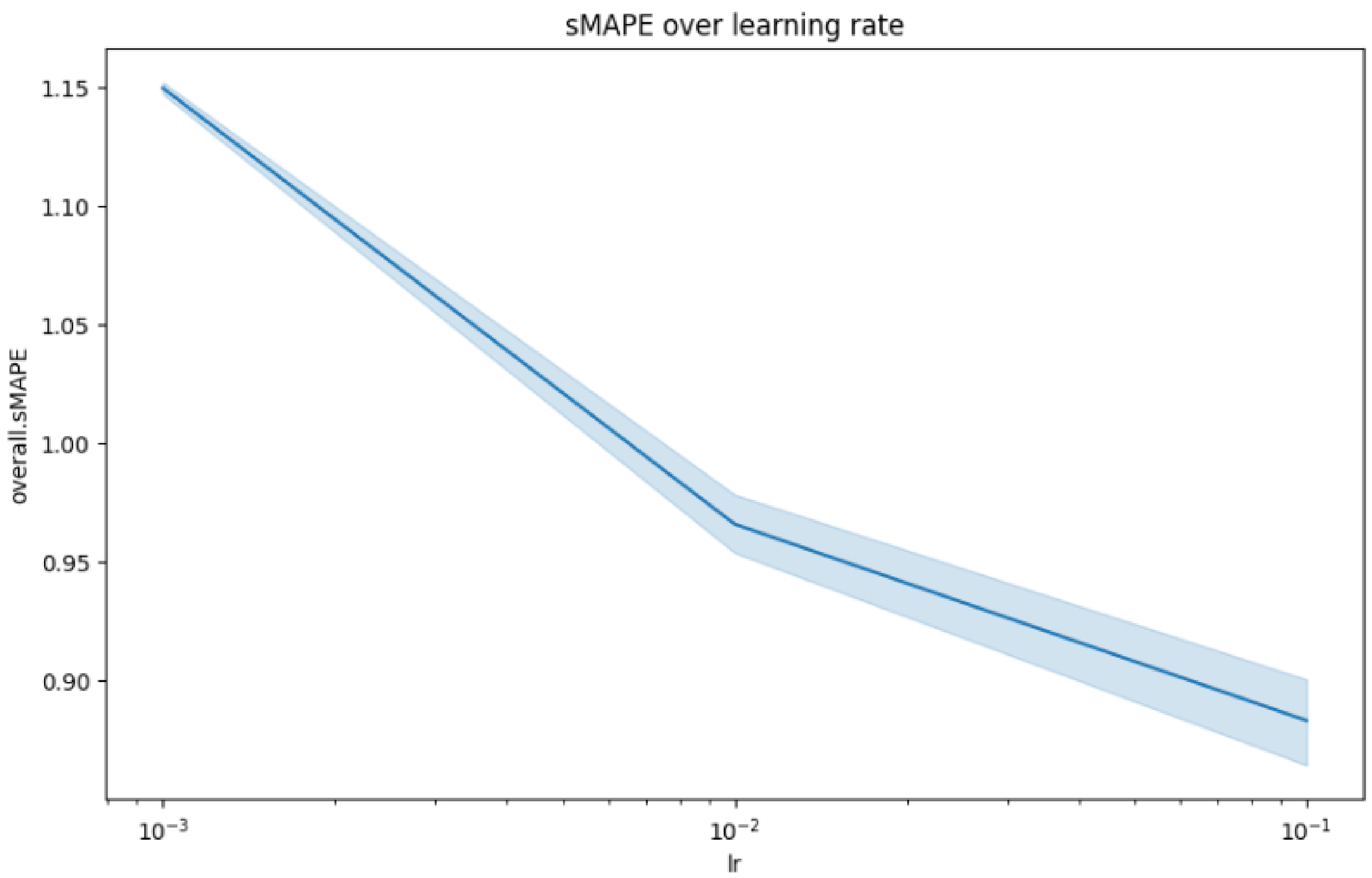
Fruthermore, we only impute on features that have a rate of missingness below 30% across the data set. After applying this feature selection 50% of the data rows have a missingness of below 10%.

Results and Insight

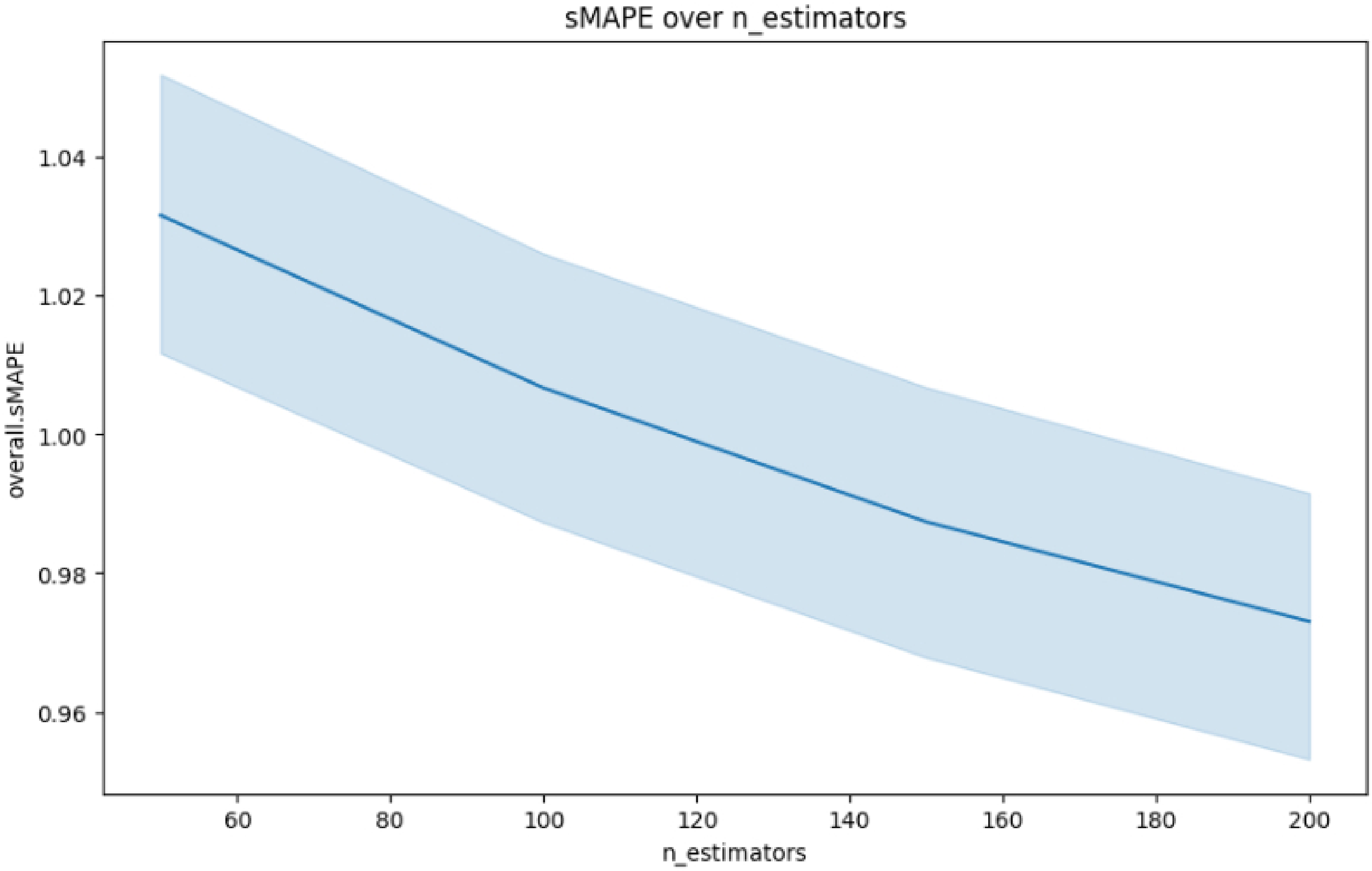
The plot shows all the different configurations tested across all difficulty levels. The results reveal a strong effect of the learning rate and the number of estimators on the performance.



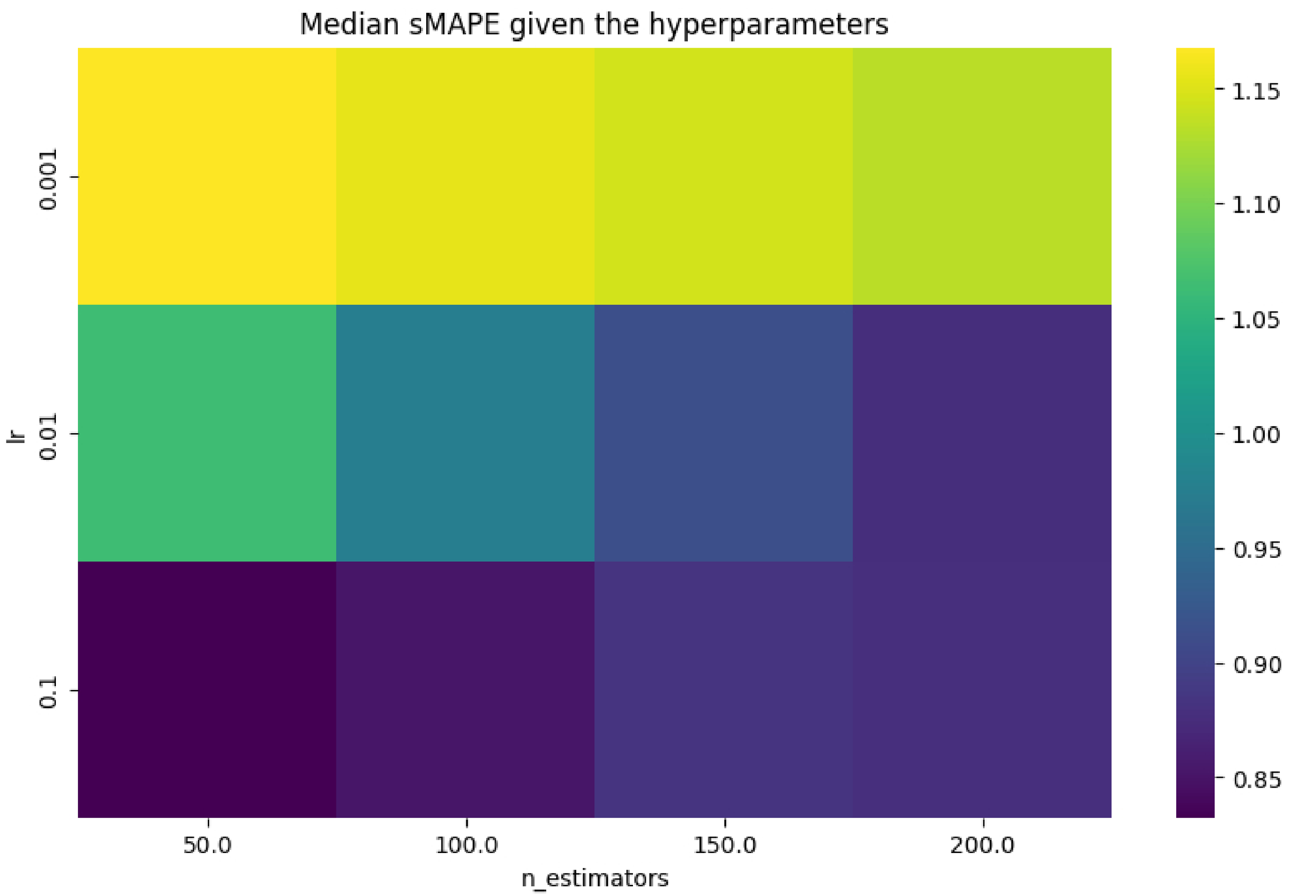
Learning rate seems to be the most influential factor. The higher the learning rate the better the prediction.



Furthermore, the smaller the amount of estimators, the better the performance.



This figure reavels the optimal configuration



Decision

Update : 09.01.2024

The results show a clear pattern, when it comes to the selection of the best hyper-parameters for the overall experiment. Therefore, we decide to rtun the experiment with a learning rate of 0.1 and 50 estimators.