

Vehicle Leasing Price Predictor

A Machine Learning approach



Authors:

- Tobias Ponesch
- Sina Haghgoo
- Finnian John Dempsey
- Vinicius Wolff
- Adrian Lehrner
- Mario Dangev

Coaches:

- Thomas Dornigg
- Jannik Neuböck

Advising Professor:

- Ronald Hochreiter

We are immensely grateful for the collaboration and support of zeb Consulting, which has greatly contributed to the success of this project.

Introduction

This project aims to develop a machine learning model to predict the leasing prices of vehicles based on various [attributes](#).

In the current macroeconomic environment, accurately forecasting leasing asset values and pricing is crucial for leasing banks. Additionally, the automotive market has experienced significant price fluctuations and supply chain disruptions, further emphasizing the need for reliable predictions. Leveraging a dataset provided by a leasing bank, our research and development efforts focus on building the most accurate prediction models.

The final outcome will be a graphical user interface (GUI) that allows users to input vehicle details and obtain leasing rate predictions. By employing state-of-the-art machine learning techniques and addressing challenges such as data quality and model selection, this project aims to provide an effective tool for leasing banks in assessing asset values and making informed pricing decisions.

Table of Contents

1. [Import Libraries](#)
2. [Computational effort](#)

3. Dataset import
4. Basic Preprocessing
5. Explanatory Data Analysis
 - 5.1 The Target variable
 - 5.2 Numerical Features
 - 5.2.1 Skewness of numerical variables
 - 5.3 Categorical Features
 - 5.4 Target variable vs. categorical features
 - 5.5 Heatmap (Correlations)
 - 5.5.1 Dropping kilowatts
6. Preprocessing and Feature Engineering
 - 6.1 Missing Values
 - 6.2 Cardinality of non-numeric features
 - 6.3 Problems with splitting -- STILL OPEN
 - 6.4 Out of Sample split
 - 6.5 Train and Test split
 - 6.6 Transformer Pipelines
 - 6.6.1 Challenge: Encoding
7. Machine Learning Modeling
 - 7.1 Choosing appropriate metric & customization approach
 - 7.2 Decision Tree
 - 7.3 Random Forest
 - 7.4 K-nearest neighbor
 - 7.5 XGBoost
 - 7.6 Support-Vector-Machine (SVM)
 - 7.7 AdaBoost Regressor
8. Test Data Performance
 - 8.1 Metrics comparison
 - 8.2 Predicted vs actual plots
9. Out of sample performance
 - 9.1 Metrics comparison
 - 9.2 Predicted vs actual plots
10. Feature Importance Analysis
 - 10.1 Decision tree feature importance
 - 10.2 Random forest feature importance
 - 10.3 XGB feature importance
 - 10.4 AdaBoost feature importance
11. Safe models
12. Debugging library versions

References

Appendix

- A1 [Encoding differences](#)
- A2 [Bar plots Test performance](#)
- A3 [Bar plots Out of sample performance](#)
- A4 [Histogram of residuals Test performance](#)
- A5 [Histogram of residuals Out of sample performance](#)
- A6 [Light models \(reduced complexity\)](#)

1 Import Libraries

To develop high-quality machine learning algorithms and streamline data processing, we utilized state-of-the-art libraries such as pandas, numpy, and sklearn in this project. These libraries enabled us to implement advanced techniques and achieve efficient data manipulation and analysis. If you run into any problems with the imported libraries, please refer to the [Debugging library versions](#) section.

2 Computational effort

To optimize the computational and time requirements of building machine learning models, we have introduced a section that allows you to choose between computing new models or importing existing ones. Additionally, you will be prompted to evaluate your machine's performance, ranging from "Ludicrous" (highest performance) to "Low" (lowest performance). Your response will determine the number of iterations and cross-validations in the subsequent Random Search process.

Moreover, the number of threads used for model building is limited to your available threads minus two, ensuring that your machine remains usable during the process. This approach aims to strike a balance between model generation and system usability.

3 Dataset import

To import the data, please specify your data folder in the following cell.

If you want to import your models, please specify the folder, from which they shall be imported in the following cell.

If you want to use a different or new dataset, please ensure that you correctly assign the column names of your dataset in the following dictionary.

This shall insure, that without much change, the models can be rebuilt on a different dataset.

		brand	model	milage	registration	duration	gear	fee	emission	cc
0	Skoda	Octavia ŠKODA Combi Style TDI DSG	201 km	03/2023	48 Monat (anpassbar)	Automatik	574,01 €	119 g/km		
1	Volkswagen	T- Cross VW Life TSI	201 km	03/2023	48 Monat (anpassbar)	Manuelle Schaltung	382,58 €	131 g/km		
2	Seat	Ibiza Austria Edition	15.000 km	10/2022	48 Monat (anpassbar)	Manuelle Schaltung	239,62 €	120 g/km		
3	Volkswagen	Polo VW	1 km	01/2023	48 Monat (anpassbar)	Manuelle Schaltung	309,11 €	127 g/km		
4	Audi	A4 Avant 40 TDI quattro S line	105.301 km	12/2019	48 Monat (anpassbar)	Automatik	587,75 €	138 g/km		
...
19053	Seat	Ateca FR 2.0 TDI DSG 4Drive	201 km	01/2023	48 Monat (anpassbar)	Automatik	692,03 €	146 g/km		
19054	Skoda	Octavia ŠKODA Combi Style TDI DSG	201 km	03/2023	48 Monat (anpassbar)	Automatik	574,01 €	187 g/km		
19055	Audi	A4 Avant 40 TDI quattro S line	105.301 km	12/2019	48 Monat (anpassbar)	Automatik	587,75 €	143 g/km		
19056	Volkswagen	Polo VW	18.903 km	06/2020	48 Monat (anpassbar)	Manuelle Schaltung	256,33 €	40 g/km		
19057	Volkswagen	Tiguan VW Life TDI	48.000 km	09/2022	48 Monat (anpassbar)	Manuelle Schaltung	539,72 €	185 g/km		

19058 rows x 12 columns

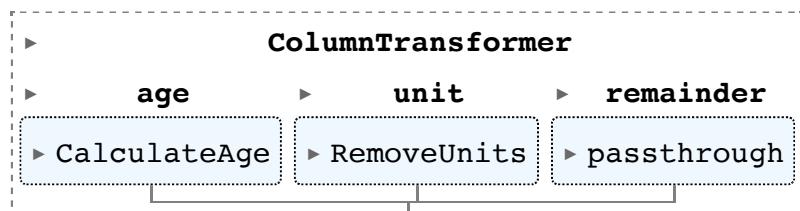
4 Basic Preprocessing

In order to ensure data usability, certain features require formatting adjustments. These adjustments involve removing units, replacing commas with decimal points, and calculating the age based on registration information. These transformations are performed within the "basic preprocessing pipeline" to prepare the data for further analysis and modeling.

None

	registration	milage	duration	fee	emission	consumption	horsepower	kilowatts
0	3	201.0	48.0	574.01	119.0	5.0	150.0	110.0
1	3	201.0	48.0	382.58	131.0	6.0	95.0	70.0
2	8	15000.0	48.0	239.62	120.0	5.0	80.0	59.0
3	5	1.0	48.0	309.11	127.0	6.0	80.0	59.0
4	42	105301.0	48.0	587.75	138.0	5.0	190.0	140.0

The shape of our basic preprocessing pipeline looks like:

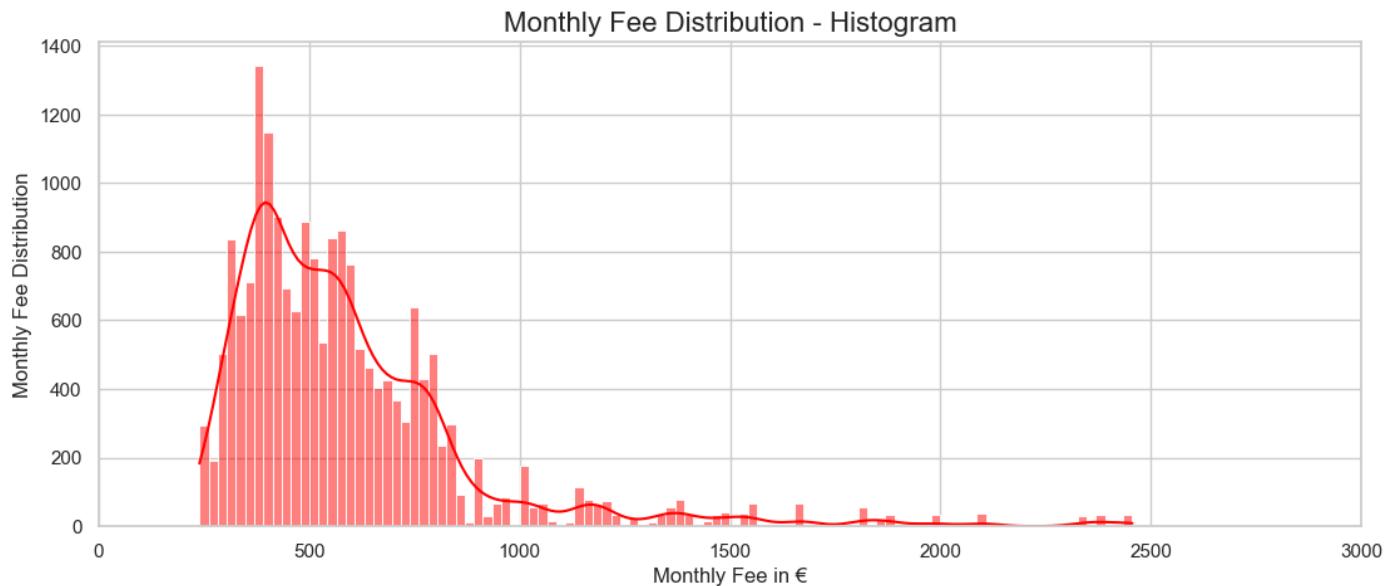


5 Explanatory Data Analysis

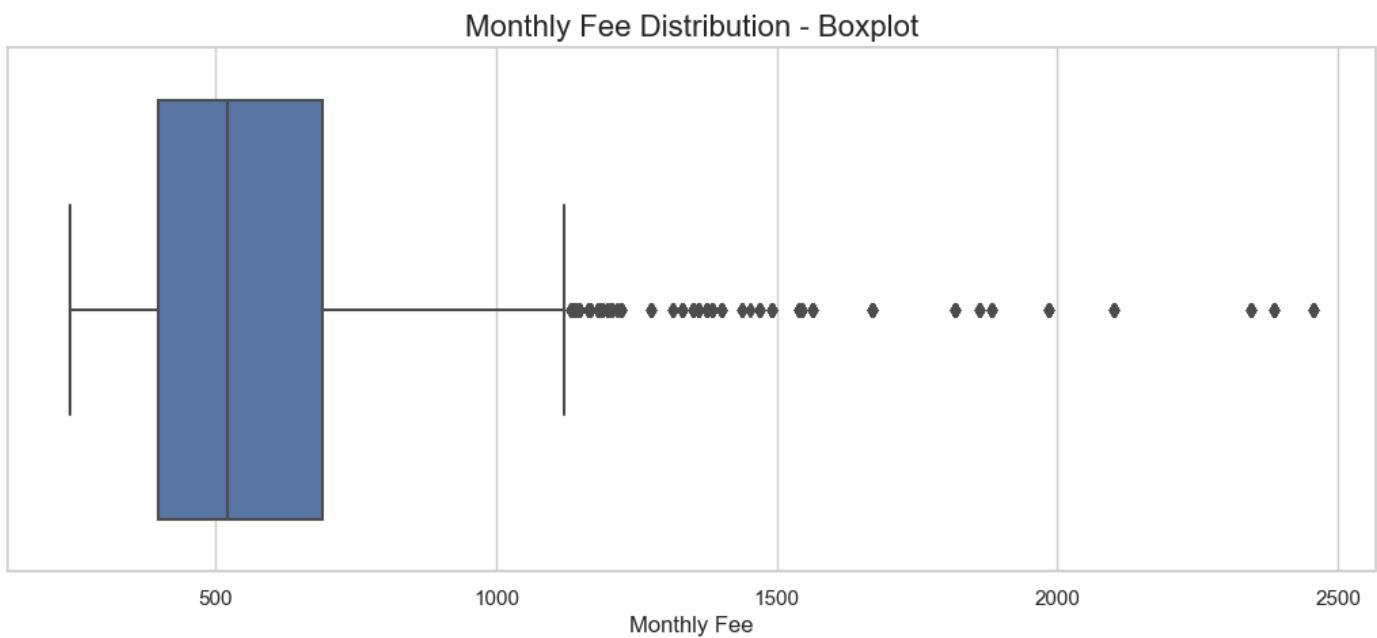
This section contains the visualization of the dataset, which serves as an essential step in understanding the underlying patterns and relationships within the data. This exploratory analysis aims to provide insights into the dataset's characteristics and unveil meaningful trends that can guide subsequent modeling efforts.

5.1 The Target variable

In this section, we will present an overview of the target variable in this analysis. Specifically, the target variable in our predictor refers to the monthly leasing price associated with a leasing asset.



On this histogram we see the monthly fee distribution for the cars in the dataset. The distribution exhibits a right-skewed pattern, indicating that a majority of the leasing rates fall towards the lower end of the scale with minimum leasing rates starting from approximately 250 euros. The peak of the distribution occurs around 350 euros, with a significant number of cars falling within this monthly fee range. Most of the values lie between 300-900 euros. What is interesting is that we can see that there are very few cars that have a leasing rate between 2000 and 2500 euros. Those cars are not outliers but rather just very expensive cars.



The boxplot for the monthly fee distribution verifies what we observed from the histogram. The minimum monthly fee value is approximately 250 euros, while the maximum reaches up to 1200 euros. The median, which represents the middle value of the distribution, lies slightly above 500 euros. Upon closer examination, it becomes apparent that the boxplot exhibits numerous outliers. However (as stated earlier), these outliers are not indicative of data anomalies but rather represent the presence of very expensive cars with exceptionally high leasing rates. This observation highlights the diversity within the dataset, as it encompasses both affordable and luxury vehicles.

5.2 Numerical Features

The numerical features gathered are:

- Mileage:

The total distance traveled by the vehicle in miles. This feature provides information about the wear and tear on the vehicle and can impact its value and leasing price.

- First Registration:

The duration, represented in months, since the vehicle's initial registration. This feature indicates the age of the vehicle and can affect its depreciation and leasing price.

- Duration:

The duration of the leasing contract in months. This feature represents the length of time for which the vehicle is leased and influences the leasing price.

- Emissions:

The amount of emissions produced by the vehicle, typically measured in grams of CO2 per kilometer. This feature provides insights into the environmental impact of the vehicle and may impact leasing decisions, especially in regions with emissions regulations.

- Consumption:

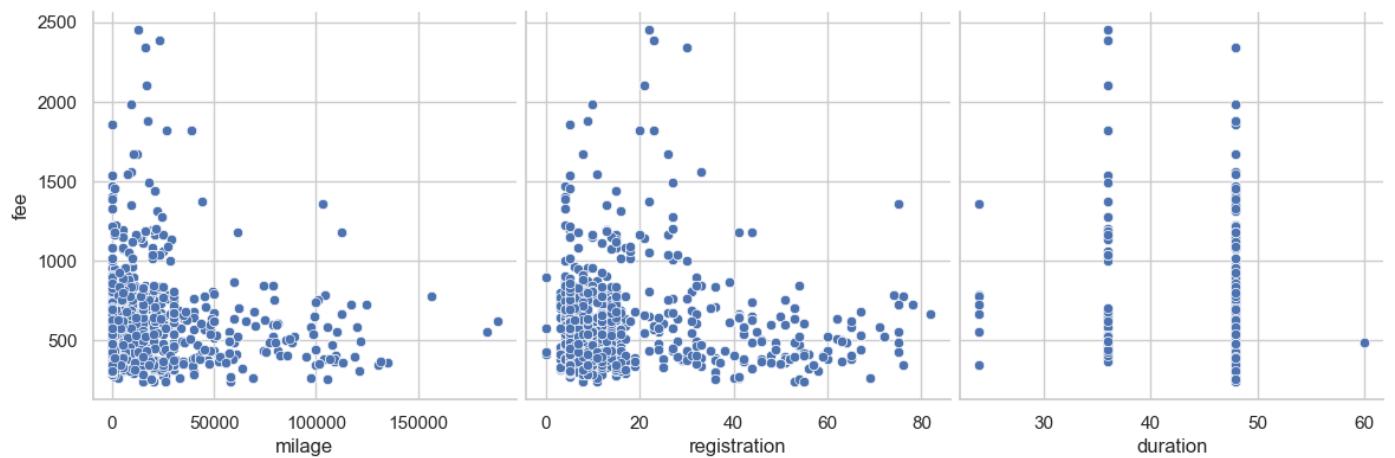
The fuel consumption of the vehicle, typically measured in liters per 100 kilometers. This feature indicates the efficiency of the vehicle in terms of fuel usage.

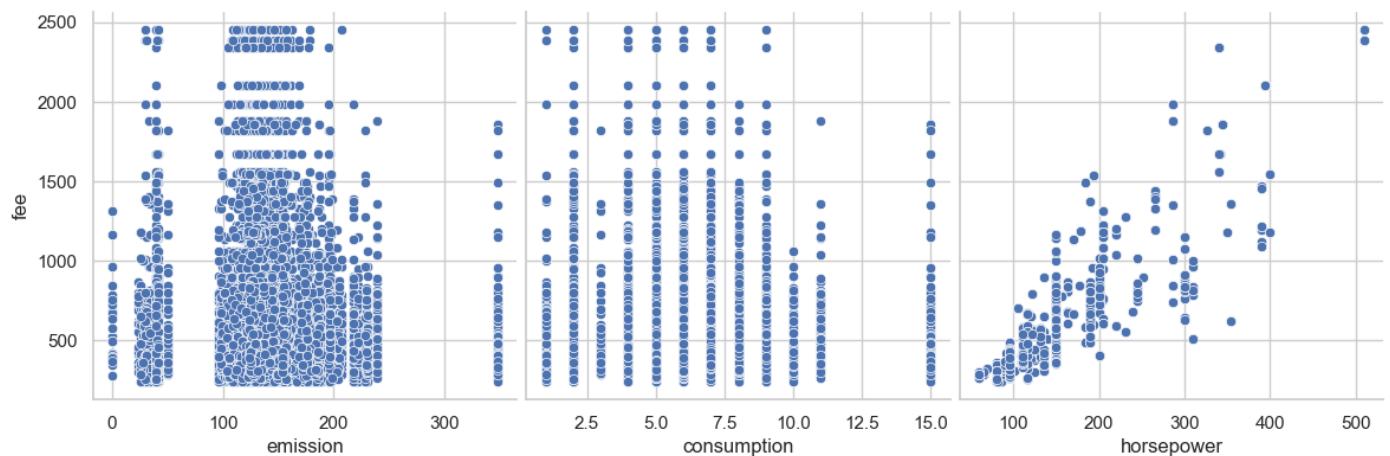
- Horsepower:

The measure of the vehicle's engine power. This feature represents the strength and performance capabilities of the vehicle.

- Kilowatts:

The power of the vehicle's engine in kilowatts. This feature provides an alternative representation of the engine power.





Scatterplots provide a visual representation to explore the relationship between numerical variables and the target variable in our dataset. In our analysis, we plotted our target variable on the y-axis against all the numeric variables on the x-axis.

From the scatterplots, several key findings emerged. First, a strong linear relationship was observed between the monthly fee and the horsepower. As the horsepower increases, the monthly fee tends to rise as well, indicating a positive correlation between these variables.

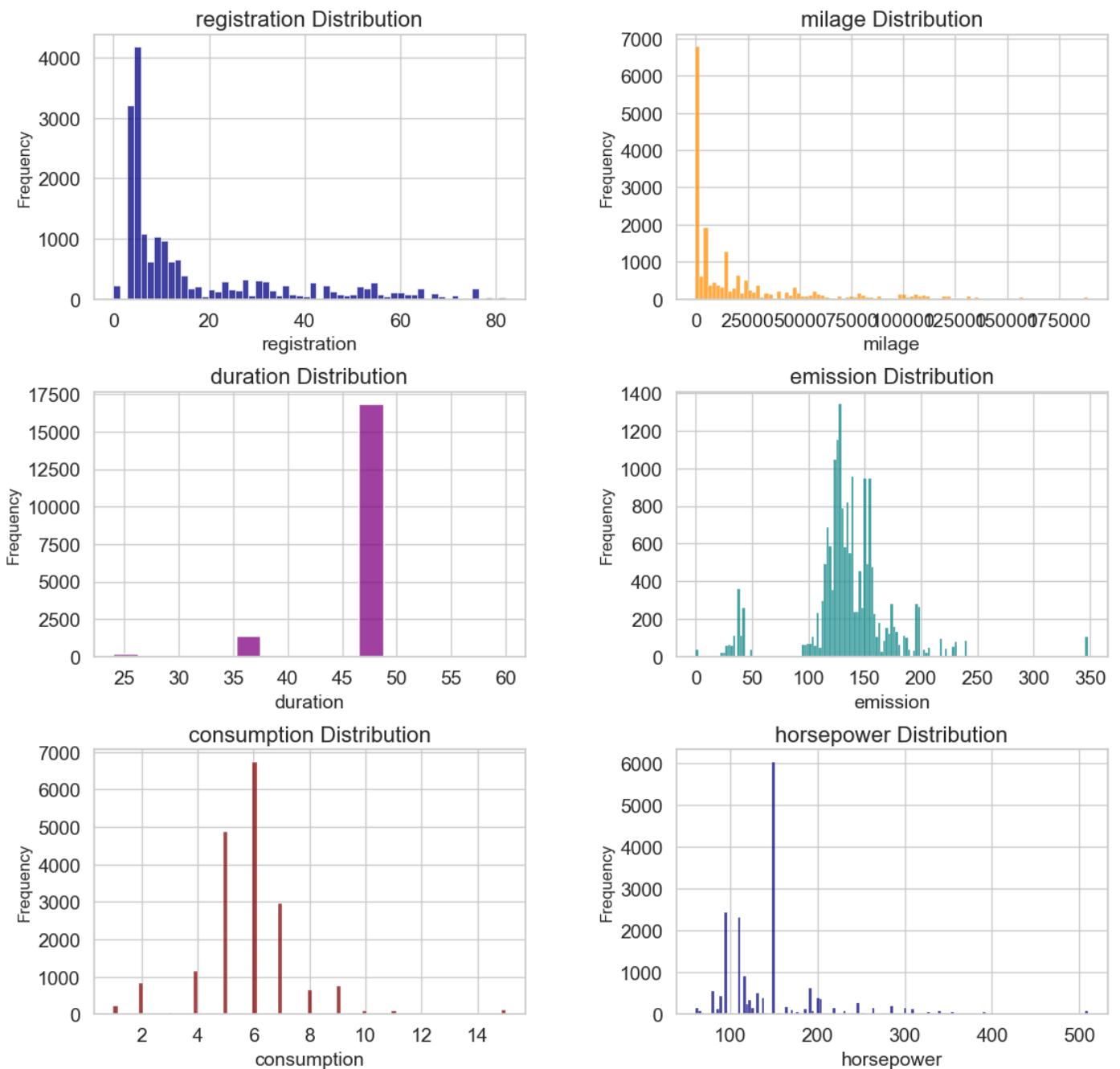
Additionally, we discovered that lower mileage and more recent initial registrations are associated with higher monthly fees. This relationship is evident from the concentrated distribution of points in the corresponding regions of the scatterplots, suggesting that these factors have a noticeable impact on leasing rates.

Examining the scatterplots for consumption and duration, we noticed an interesting pattern. Despite variations in duration and consumption, the monthly fees are relatively evenly distributed. This implies that these variables may not exert a significant influence on the monthly fee, as indicated by the consistent spread of points across different durations and consumption levels.

Regarding the emission variable, we observed a dense cluster of points in the middle range, signifying a large number of vehicles with emission values between approx. 100-250. This indicates that both low-cost and high-cost vehicles exist within this emission range, potentially reflecting a diverse market segment with various pricing factors beyond emissions alone.

Overall, these insights from the scatterplots shed light on the relationships between the numerical variables and the monthly fee, providing valuable information for feature selection and understanding the factors influencing leasing rates.

5.2.1 Skewness of numerical variables



When examining the skewedness of the numerical variables in our dataset, interesting patterns emerge. Specifically, the registration and mileage variables exhibit right-skewed distributions, indicating a high concentration of new vehicles. This suggests that a significant portion of the dataset comprises recently registered vehicles with relatively low mileage.

In contrast, the emission and consumption variables demonstrate symmetric distributions, implying a more balanced distribution of values. The absence of skewness in these variables suggests that the dataset encompasses a diverse range of emission and consumption values without a pronounced bias towards higher or lower values.

5.3 Categorical Features

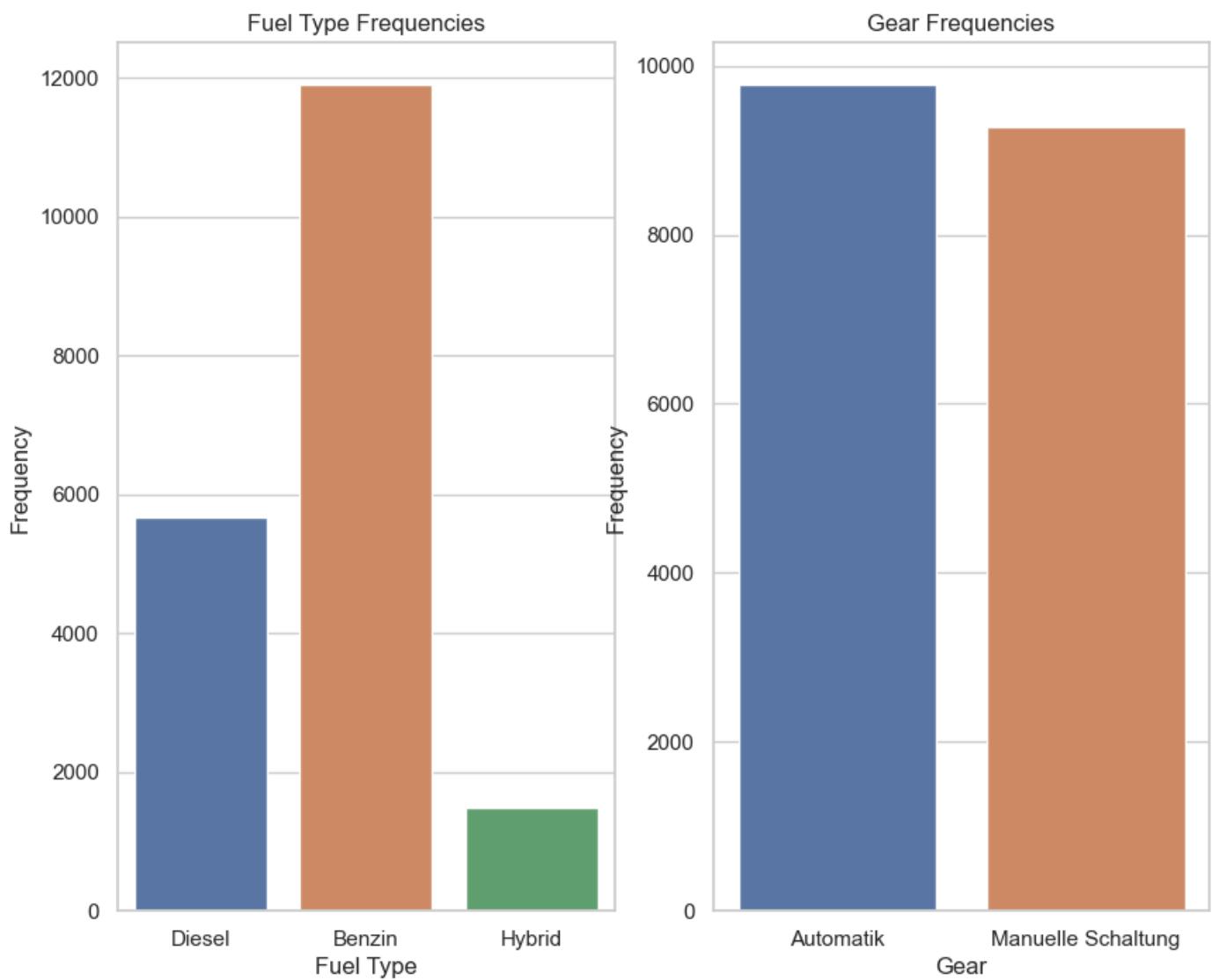
The categorical features gathered are:

- Brand:
The brand or manufacturer of the vehicle. This feature represents the specific company or brand associated with the vehicle's production.
- Model:
The specific model or variant of the vehicle. This feature provides detailed information about the vehicle's specific version or edition.
- Gear:
The type of gear or transmission system used in the vehicle. This feature indicates whether the vehicle has a manual or automatic gear system.
- Fuel:
The type of fuel used by the vehicle. This feature represents the fuel source required for the vehicle's operation, such as gasoline, diesel, or hybrid.

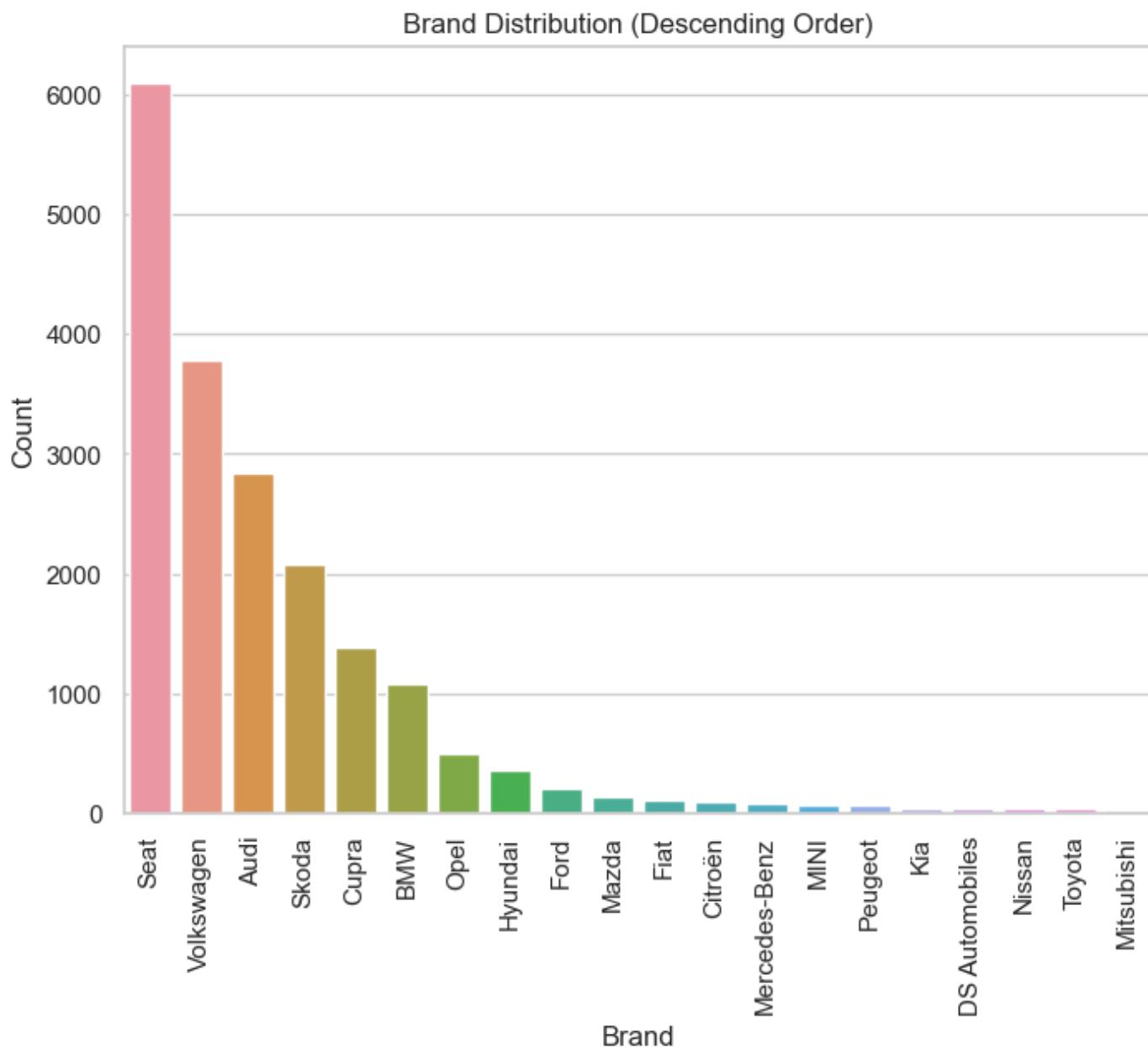
	brand	model	gear	fuel
0	Skoda	Octavia ŠKODA Combi Style TDI DSG	Automatik	Diesel
1	Volkswagen	T-Cross VW Life TSI	Manuelle Schaltung	Benzin
2	Seat	Ibiza Austria Edition	Manuelle Schaltung	Benzin
3	Volkswagen	Polo VW	Manuelle Schaltung	Benzin
4	Audi	A4 Avant 40 TDI quattro S line	Automatik	Diesel
...
19053	Seat	Ateca FR 2.0 TDI DSG 4Drive	Automatik	Diesel
19054	Skoda	Octavia ŠKODA Combi Style TDI DSG	Automatik	Diesel
19055	Audi	A4 Avant 40 TDI quattro S line	Automatik	Diesel
19056	Volkswagen	Polo VW	Manuelle Schaltung	Benzin
19057	Volkswagen	Tiguan VW Life TDI	Manuelle Schaltung	Diesel

19058 rows × 4 columns

Text(0.5, 1.0, 'Gear Frequencies')



The barplots provide insights into the fuel types and gear types of the vehicles in our dataset. We can observe that the dataset comprises three primary fuel types: diesel, gasoline, and hybrid vehicles. Additionally, the barplots reveal the presence of two gear types: automatic and manual shifts. This highlights the transmission options available within the dataset.



Upon exploring the brands present in our dataset, we identified several notable findings. The most frequently occurring brands among the vehicles in our dataset are Seat, Volkswagen, Audi, Skoda, Cupra, BMW, and Opel.

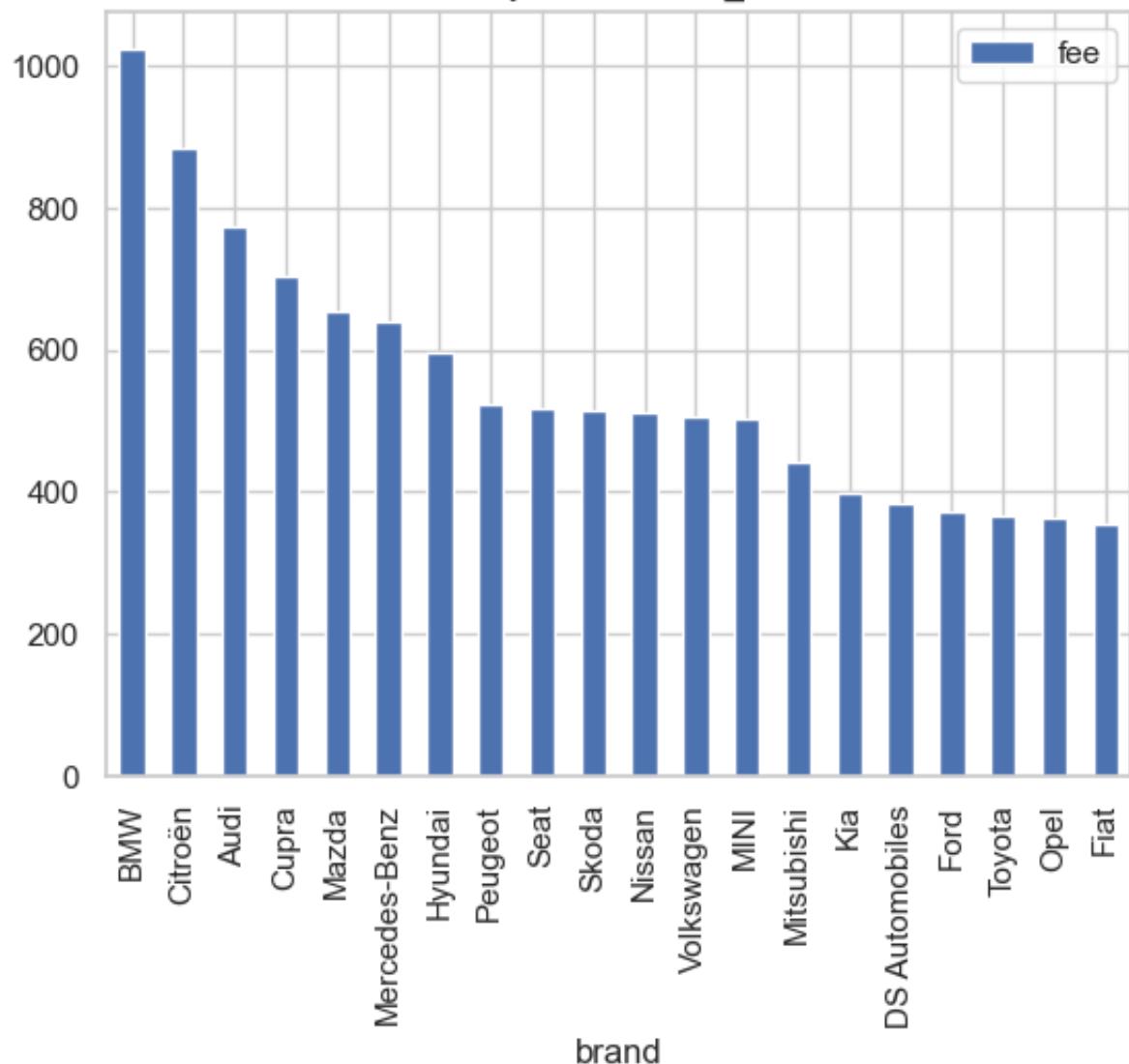
The inclusion of these brands in our dataset represents a mix of mainstream vehicles, providing a comprehensive view of the market and enabling us to analyze the impact of brand types on the monthly leasing fees.

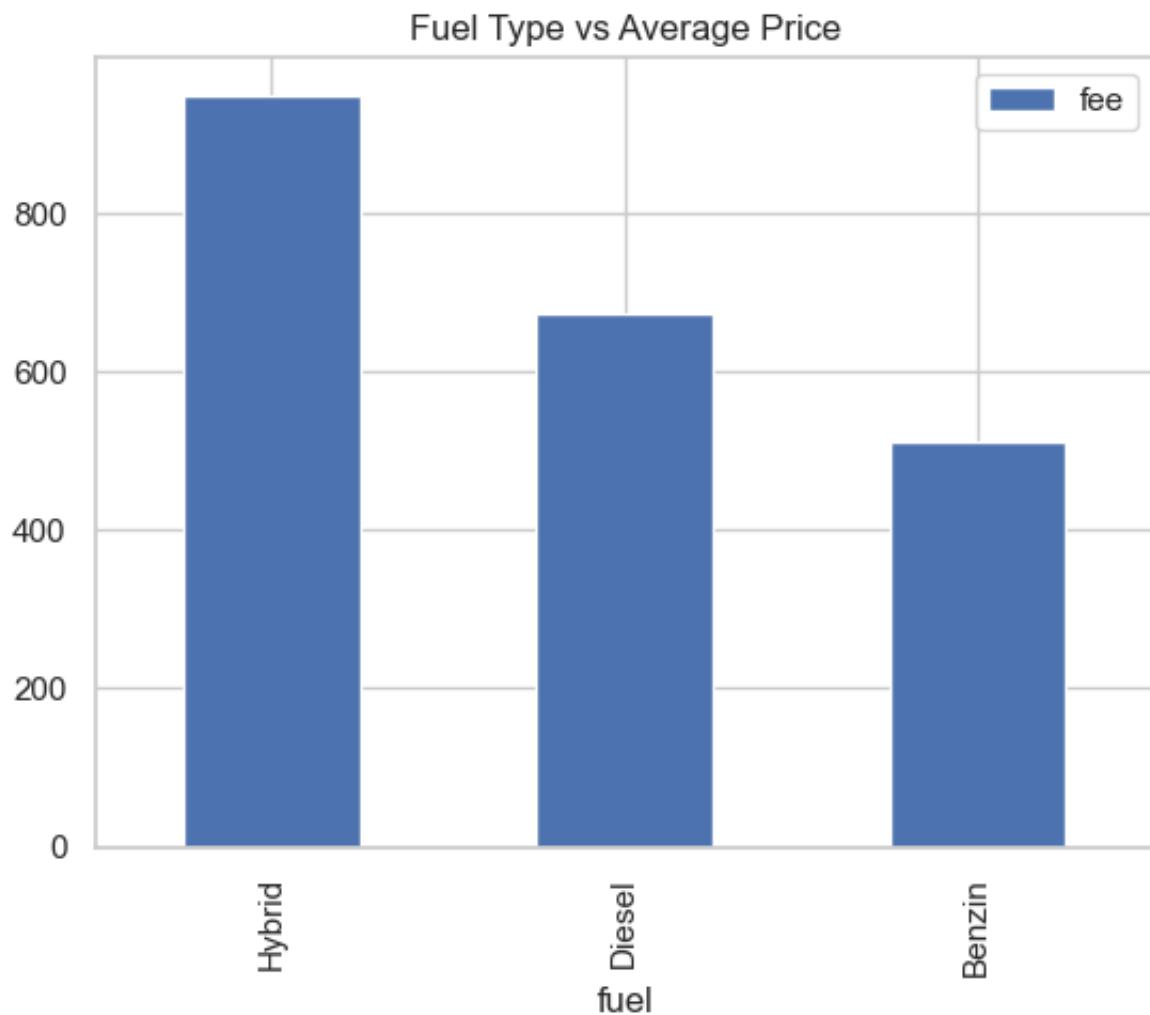
5.4 Target variable vs. categorical features

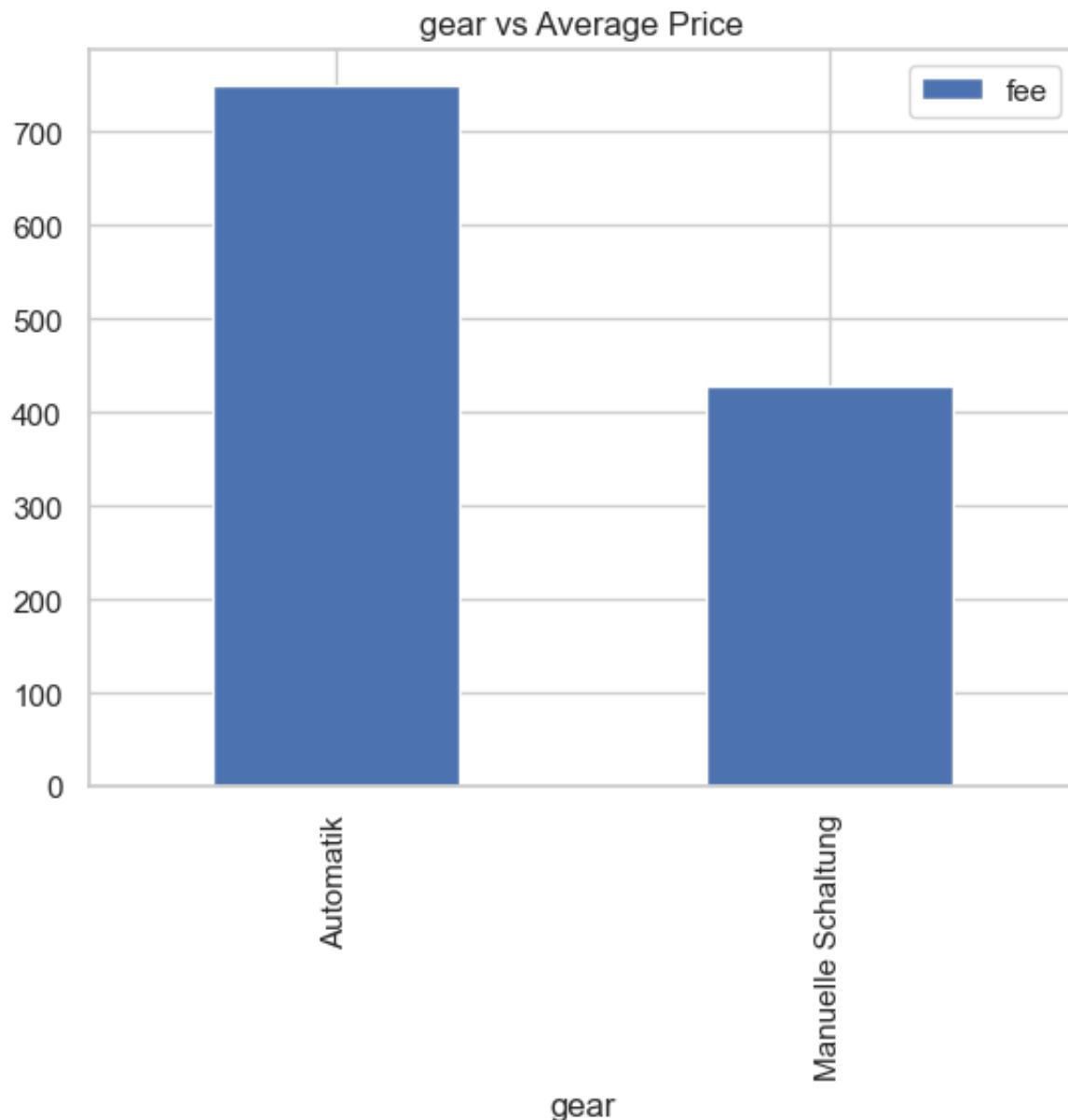
This section provides an overview of how the categorical features influence the target variable, which is the monthly fee.

<Figure size 2500x600 with 0 Axes>

Monthly fee vs brand_name







Analyzing the relationship between the target variable and the categorical variables in our dataset revealed interesting insights. Plotting the target variable against different categorical variables allowed us to examine their impact on leasing rates.

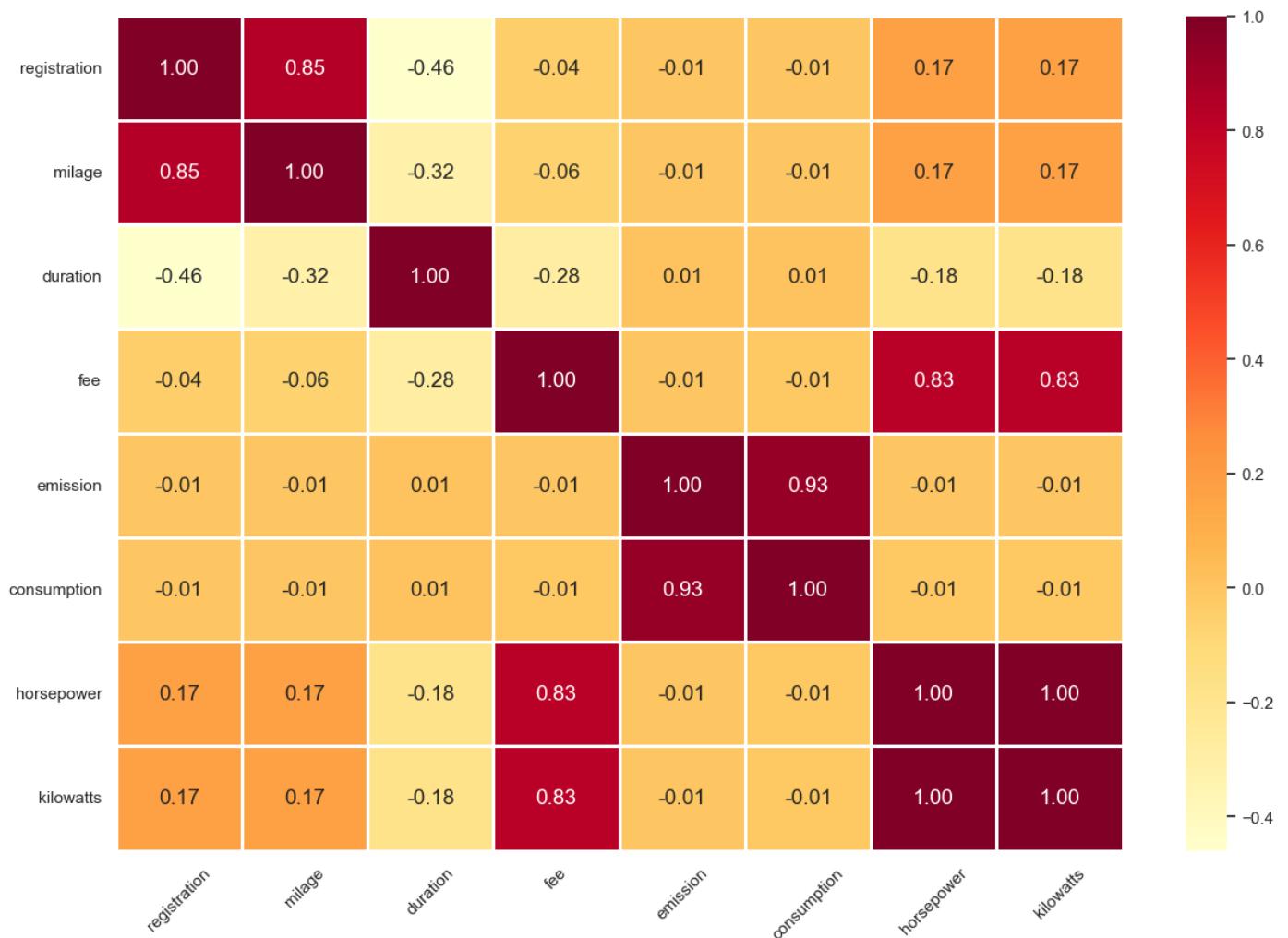
From the visualizations, we observed that BMW models, Citroën, and Audi tend to have higher average leasing prices compared to other brands.

Moreover, when considering the fuel type, we found that hybrid vehicles tend to have higher monthly fees compared to gasoline or diesel vehicles. This is attributed to the increased cost of hybrid technology and the potential for fuel savings over time.

Additionally, we observed that automatic cars are generally more expensive to lease compared to cars with manual shifting. This is very likely perceived convenience and comfort associated with automatic transmission, which can contribute to higher demand and pricing.

5.5 Heatmap (Correlations)

In this section, we present a plot showcasing the correlations between the features in the dataset. By visualizing the correlations, we gain insights into the relationships and dependencies among the different attributes. This correlation plot provides a comprehensive overview of the interplay between the numerical and categorical features, highlighting which variables are positively or negatively correlated. Understanding these correlations helps us identify potential patterns and dependencies that can significantly impact the target variable, providing valuable insights for subsequent analysis and model development.



This correlation matrix measures the relationships between different variables related to a car, namely: milage, first_registration, duration, monthly_fee, emission_value, consumption, horsepower, and kilowatts. The correlations range from -1 (perfect negative correlation, as one variable increases, the other decreases) through 0 (no correlation, the variables do not move together) to +1 (perfect positive correlation, the variables increase and decrease together).

- **Monthly_fee and Horsepower/Kilowatts:**

These pairs show very strong positive correlations (0.827053 and 0.826905, respectively).

This suggests that cars with higher horsepower or kilowatts are associated with higher monthly fees. This could mean that more powerful vehicles tend to have higher monthly costs, potentially due to reasons such as higher insurance premiums, increased fuel consumption, or greater maintenance requirements.

- **Monthly_fee and Duration:**

There is a moderate negative correlation (-0.280965) between these variables, indicating that as the duration of ownership increases, the monthly fee decreases. This could be because the costs associated with a car (like loan payments or certain insurance costs) often decrease over time.

- **Monthly_fee and Mileage/First_registration:**

The correlation between monthly_fee and these two variables is relatively weak (-0.060930 and -0.041417, respectively). This suggests that the monthly fee does not change significantly with changes in the mileage of the car or its first registration date. It's worth noting though, that in some cases, older cars (with an earlier first registration date) or cars with higher mileage could potentially have higher maintenance costs which could affect the monthly fee.

- **Monthly_fee and Emission_value/Consumption:**

The correlations here are also very weak (-0.008253 and -0.012807, respectively). These small negative correlations suggest that cars with higher emissions or consumption are associated with slightly lower monthly fees, although this relationship is very weak. This may be because cars with higher emissions or fuel consumption tend to be older models, which could have lower associated costs in some areas (like lower insurance or depreciated value).

- **Mileage and First_registration:**

These two variables show a strong positive correlation of 0.845908, implying that as the age of the car (as suggested by the first registration) increases, so does the mileage it has run. This is a fairly intuitive relationship since older cars have typically been driven more.

- **Duration and First_registration:**

These two variables have a moderate negative correlation of -0.459295, indicating that as the duration of ownership increases, the car tends to be newer (i.e., has a later first registration date). This might imply that people tend to keep newer cars for longer periods.

Our analysis of the correlation matrix reveals two key variables that significantly influence the monthly fee, namely horsepower and leasing duration. Horsepower shows a strong positive correlation, suggesting that more powerful cars generally incur higher monthly fees. Conversely, duration of ownership has a negative correlation with the monthly fee, indicating that longer leasing contract durations result in lower monthly fees. Other variables, including mileage, first_registration, emission_value, and consumption, show a weaker correlation with the monthly_fee, suggesting a lesser direct impact on this target variable.

As expected, horsepower and kilowatts show a perfect correlation, because they are different units of the same attribute of a car, the engine power. To prevent multicollinearity, which can complicate interpretation of the model, we decided to exclude kilowatts from our subsequent models. This decision helps to streamline our model by eliminating redundant information, focusing on the most relevant predictors for the monthly fee.

5.5.1 Dropping kilowatts

As anticipated, there is a perfect 1:1 correlation between kilowatts (kW) and horsepower (HP), where the relationship is defined as $P[\text{kW}] = 0.7457 * P[\text{HP}]$.

Given this direct correlation, we decided to exclude the kilowatts feature from the dataset when constructing our models.

6. Preprocessing and Feature Engineering

In this section, we delve into the preprocessing and feature engineering steps of our notebook. These crucial steps lay the foundation for preparing the data and optimizing its suitability for the machine learning process. We begin by splitting the data into out-of-sample, test, and train sets to ensure robust model evaluation and prevent overfitting.

Once the data is appropriately split, we proceed to define transformer pipelines using the renowned scikit-learn library. These pipelines facilitate systematic data transformations and feature engineering, ensuring consistency and efficiency throughout the machine learning process. By employing transformer pipelines, we can seamlessly apply various preprocessing techniques such as scaling, encoding categorical variables, handling missing values, and creating interaction features.

By carefully designing and implementing these preprocessing and feature engineering steps, we enhance the quality and representativeness of our data, enabling the machine learning models to capture meaningful patterns and make accurate predictions. The transformer pipelines in scikit-learn provide a flexible and comprehensive framework for streamlining these essential data preparation tasks, promoting reproducibility and scalability in our analysis.

6.1 Missing Values

Only two features present missing values, which are the emission and the consumption.

Summary of Missing Values:

```
registration      0
milage          0
duration        0
fee             0
emission       612
consumption    612
horsepower     0
brand           0
model           0
gear            0
fuel            0
dtype: int64
```

Missing values will be imputed in the transformer pipeline, using the standard imputed provided by the scikit-learn package.

6.2 Cardinality of non-numeric features

Analyzing the cardinality helps us understand the number of distinct categories within each feature. In this case, the "Brand" feature consists of 20 unique brands, indicating a moderate level of variation. On the other hand, the "Model" feature has a higher cardinality with 346 unique models, suggesting a more diverse range of vehicle variations.

The "Gear" feature has only two categories, indicating a binary classification of the transmission type (e.g., manual vs. automatic). Similarly, the "Fuel" feature has three categories representing different fuel types (e.g., gasoline, diesel, hybrid).

Understanding the cardinality of categorical features is important for various aspects of data analysis, including feature selection, encoding strategies, and model interpretation. High cardinality may require careful handling to avoid overfitting or computational challenges, while low cardinality features can simplify modeling and analysis tasks.

The cardinality of the model feature was a major concern in [encoding categorical features](#). The high cardinality of the "model" feature lead to high dimensionality of the dataframe and the models.

```
brand      20
model     346
gear       2
fuel       3
dtype: int64
```

6.3 Problems with splitting

During the dataset splitting process using the "train_test_split" function, stratification is required to ensure that one-hot encoding works properly. However, a challenge arises when dealing with entries in the "model" column that appear only once or twice. Stratification cannot be applied to these unique or minimally occurring entries.

To address this issue, we have identified three possible approaches. The first approach involves dropping the single or double occurrence entries of the "model" column. This reduces the complexity introduced by the limited occurrences during stratification. Alternatively, the second approach suggests duplicating or tripling the once or twice occurring entries to increase their representation in the dataset. This approach helps maintain balance and avoids losing potentially valuable information. Another option is to create a combined category for these specific models, treating them as a separate group during the splitting process. This approach can preserve the uniqueness of these entries while ensuring proper stratification.

For the current implementation, we have decided to drop the entries that appear only once or twice. However, we acknowledge that other approaches may be explored in the future to fully utilize the data from these unique or minimally occurring entries.

Models dropped: 19

This section was significant for the prior encoding method, OneHot encoding, but would be redundant now. However, to be able to use old models, which were built using OneHot encoding, we will keep this part.

6.4 Out of Sample split

```
Size of the sample data: (16183, 11) with a mean of: 593.0066569857258
Size of out of sample data: (2856, 11) with a mean of: 592.3941316526611
```

6.5 Train and Test split

```
Size of the train data: (12137, 10) with a mean fee of: 593.1457435939689
Size of the test data: (4046, 10) with a mean fee of: 592.5894315373208
```

6.6 Transformer Pipelines

In this section, we focus on the implementation of transformer pipelines, which play a vital role in the preprocessing and feature engineering stages of our machine learning workflow.

6.6.1 Challenge: Encoding

When it comes to choosing between OneHot Encoder, Label Encoder, and Ordinal Encoder, the

decision depends on the specific requirements of the machine learning models being used. Let's evaluate the applicability of these encoders for different models:

Decision Tree and Random Forest:

Decision trees and random forests can handle both categorical and numerical features effectively. They are not influenced by the encoding technique used, making them compatible with all three encoders. OneHot Encoder is suitable for decision trees and random forests as it can represent categorical variables without imposing an ordinal relationship. Label Encoder and Ordinal Encoder can also be used, but they might introduce an implicit order that may or may not be appropriate for the model.

XGBoost and AdaBoost:

XGBoost and AdaBoost are ensemble learning methods that use gradient boosting. Similar to decision trees and random forests, they can handle both categorical and numerical features. OneHot Encoder, Label Encoder, and Ordinal Encoder can be used with both models. OneHot Encoder may result in high dimensionality, but XGBoost's ability to handle sparse data makes it feasible. However, considering the potential memory and computational limitations, careful consideration should be given to the choice of encoding.

KNN (K-Nearest Neighbors):

KNN is a distance-based algorithm that calculates similarity between data points. OneHot Encoder is not suitable for KNN as it can lead to the curse of dimensionality due to the high dimensionality introduced. Label Encoder and Ordinal Encoder can be used for KNN, but they assume an underlying order or ranking that may not be appropriate for categorical features. Thus, it is advisable to use alternative encoding techniques such as target-based encoding or frequency encoding to preserve the categorical information while mitigating the dimensionality issue.

SVR (Support Vector Regression):

SVR is a regression method that uses support vectors to find the best fit. Similar to KNN, OneHot Encoder can result in high dimensionality and is not recommended for SVR. Label Encoder and Ordinal Encoder can be used with SVR, but they assume an ordinal relationship that may not be valid for categorical features. Alternative encoding methods such as target encoding or effect encoding might be more suitable for SVR to capture the impact of categorical features accurately.

In summary, the choice of encoder depends on the specific machine learning models used. OneHot Encoder is generally suitable for decision trees, random forests, and XGBoost, while Label Encoder and Ordinal Encoder may introduce implicit ordering that can be inappropriate for some models. KNN and SVR require careful consideration of encoding techniques to address high dimensionality and preserve the meaningful information within categorical features.

Sklearn's label encoder is used for the target variable, not for feature variables. Ordinal encoding

is supposed to be used on categorical feature variables. Ordinal encoding is easier to use than writing a custom encoder using label encoding. Ordinal implies an underlying rank of values, although there might be no real underlying ranking. Ordinal encoding ranks alphabetically, which might not make sense.

SHAP values for OneHot encoded models can be aggregated by summing the individual SHAP values, according to: <https://github.com/slundberg/shap/issues/397>

We trained all models with OneHot encoding and with ordinal encoding and recognized, that there is little difference in most of their prediction performances.

Even KNN, which could be effected by the introduced ranking of the ordinal encoder, showed similar performance to the KNN model with OneHot encoding.

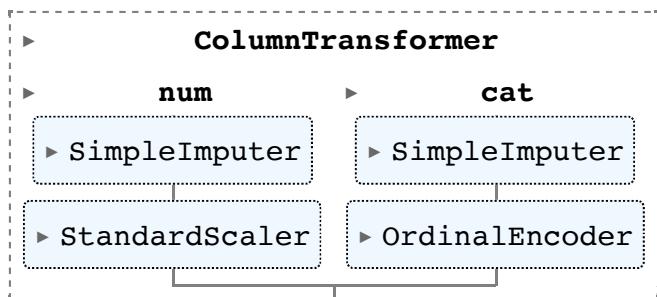
As expected, the ordinal encoding influenced the SVR model's performance immensely. We assume that this is due to the introduced ranking in the ordinal encoding.

We additionally introduced AdaBoost to replace the SVR model for Ordinal encoding, because the SVR model was very much effected by the differences in encoding.

As expected, computation times were greatly reduced with the introduction of ordinal encoding.

For evaluation of the differences of OneHot encoding and Ordinal encoding, refer to the [Appendix](#)

Ordinal encoding is being used.



7 Machine Learning Modeling

7.1 Choosing appropriate metric & customization approach

Choosing the appropriate evaluation metrics is a critical step in assessing the performance of machine learning models. These metrics help us understand how well our models are performing and compare different models or algorithms against each other. In the context of our project, where we are predicting monthly leasing prices, we have selected several evaluation metrics to evaluate the quality of our models.

The Mean Squared Error (MSE) is a widely used metric that calculates the average squared difference between the predicted and actual values. It provides a measure of how close our predictions are to the true values, with lower values indicating better performance. The Root Mean Squared Error (RMSE) is derived from MSE by taking the square root of the average squared difference, which provides a more interpretable metric in the original scale of the target variable.

The Mean Absolute Error (MAE) is another commonly used metric that calculates the average absolute difference between the predicted and actual values. Like MSE, lower values of MAE indicate better model performance. MAE is less sensitive to outliers compared to MSE, making it a suitable choice when extreme values are present in the data.

The R-squared (R^2) metric measures the proportion of variance in the target variable that is explained by the model. It ranges between 0 and 1, with higher values indicating a better fit. R^2 is a valuable metric for assessing the overall goodness-of-fit of the model.

Additionally, we have chosen the Mean Absolute Percentage Error (MAPE or MAPR) as an evaluation metric. MAPE calculates the average percentage difference between the predicted and actual values, providing insights into the relative magnitude of errors. MAPE or MAPR is useful when we want to understand the accuracy of our predictions in relation to the actual values.

By utilizing these evaluation metrics, we can comprehensively evaluate the performance of our models and gain insights into their accuracy, precision, and generalization capabilities. This enables us to make informed decisions regarding model selection and fine-tuning to improve the predictive capabilities of our system.

For evaluation on the train and test set, we chose the following measurements:

- MSE: Mean-Square-Error
- RMSE: Root-Mean-Square-Error (for easier interpretation)
- MAE: Mean-Absolute-Error
- R^2
- MAPR: Mean Absolute Percentage Residual(Error)

Using a scoring dictionary, the RandomSearch algorithm calculates all scoring values, although it only refits on the given "refit" parameter, for which we used MSE. We also tried MAE and MAPR, but this did not make a noteable difference.

7.2 Decision Tree

In this section, we focus on building a decision tree model for predicting vehicle leasing prices. Decision trees are powerful machine learning algorithms that can effectively handle both numerical and categorical data. They provide interpretable models that mimic the decision-making process, making them widely used and easily understandable.

To construct the decision tree model, we define a parameter distribution that includes various hyperparameters such as 'min_samples_split', 'min_samples_leaf', 'ccp_alpha', and 'random_state'. These hyperparameters control the behavior and complexity of the decision tree and need to be optimized to achieve the best performance.

We create a pipeline that includes a [preprocessor](#) for data transformation and a `DecisionTreeRegressor` as the main model. The [preprocessor](#) ensures that the data is properly prepared before being fed into the decision tree model.

To find the optimal combination of hyperparameters, we perform a randomized search with cross-validation using `RandomizedSearchCV`. This technique allows us to efficiently explore different hyperparameter settings and evaluate their impact on model performance.

After fitting the decision tree model to the training data, we evaluate its performance using various metrics on both the training and test sets. These metrics provide insights into the model's accuracy, precision, and generalization capabilities.

Additionally, we analyze the best hyperparameter values obtained from the randomized search and present them in a `DataFrame`. This information helps us understand the configuration of the decision tree model that yielded the best results.

The following section defines the final `Decision Tree Regressor`, or imports an existing one:

Evaluation Metrics:

	Decision Tree Train	Decision Tree Test
MSE	46.588577	93.093921
RMSE	6.825583	9.648519
MAE	2.636798	3.217996
R2	0.999496	0.998996
MAPR	0.005080	0.006203

```
{'ccp_alpha': 0.05648616489184735, 'criterion': 'squared_error', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 2, 'min_samples_split': 20, 'min_weight_fraction_leaf': 0.0, 'random_state': 2023, 'splitter': 'best'}
```

7.3 Random Forest

In this section, we focus on building a random forest model for predicting vehicle leasing prices. Random forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is known for its robustness, ability to handle complex data, and resistance to overfitting.

We define a parameter distribution that includes hyperparameters such as 'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'random_state'. These hyperparameters control the behavior and complexity of the random forest model.

By performing a randomized search with cross-validation using RandomizedSearchCV, we explore different hyperparameter settings and evaluate their impact on model performance.

After fitting the random forest model to the training data, we evaluate its performance using various metrics on both the training and test sets. This helps us assess the model's accuracy and generalization capabilities.

The following section defines the final Random Forest Regressor, or imports an existing one:

Evaluation Metrics:

	Random Forest Train	Random Forest Test
MSE	42.209474	74.309919
RMSE	6.496882	8.620320
MAE	2.325318	3.005066
R2	0.999543	0.999199
MAPR	0.004366	0.005742

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': 20, 'max_features': 1.0, 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 2, 'min_samples_split': 10, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 550, 'n_jobs': None, 'oob_score': False, 'random_state': 2023, 'verbose': 0, 'warm_start': False}
```

7.4 K-nearest neighbor

In this section, we focus on building a K-Nearest Neighbors (KNN) model for predicting vehicle leasing prices. KNN is a non-parametric algorithm that uses the nearest neighbors in the training data to make predictions. It is known for its simplicity and versatility in handling different types of data.

We define a KNN pipeline that includes a preprocessor for data transformation and a KNeighborsRegressor as the main model.

To find the optimal combination of hyperparameters, we perform a randomized search with cross-validation using RandomizedSearchCV. The hyperparameters include 'n_neighbors', 'leaf_size', 'weights', and 'p', which control the number of neighbors, the leaf size of the tree, the weight function used in predictions, and the distance metric, respectively.

Evaluation Metrics:

	KNN Train	KNN Test
MSE	11.761946	336.703067
RMSE	3.429569	18.349470
MAE	0.702970	8.051065
R2	0.999873	0.996370
MAPR	0.001378	0.014772

```
{"algorithm": "auto", "leaf_size": 82, "metric": "minkowski", "metric_params": None, "n_jobs": None, "n_neighbors": 6, "p": 1, "weights": "distance"}
```

Despite the fact that KNN should theoretically consider the introduced ranking due to its reliance on distance-based neighbor grouping, it exhibited similar performance to the OneHot encoded KNN model. This suggests that the impact of the ranking on the KNN algorithm's predictions may be minimal in this particular context.

7.5 XGBoost

In this section, we focus on building a regression model using XGBoost (Extreme Gradient Boosting). XGBoost is a powerful machine learning algorithm known for its exceptional performance in various domains. It is an ensemble learning method that combines multiple decision trees to make accurate predictions. XGBoost incorporates gradient boosting techniques and introduces additional regularization to enhance model generalization and handle complex data patterns effectively.

To build the XGBoost model, we utilize a RandomizedSearchCV approach to search for the optimal combination of hyperparameters. These hyperparameters include the maximum depth of the trees, learning rate, number of estimators, gamma, subsample, colsample_bytree, min_child_weight, reg_lambda, reg_alpha, tree_method, and random_state. By performing cross-validation during the search, we ensure robust model evaluation and selection of hyperparameters that yield the best performance.

In addition XGBoost offers **GPU acceleration**, which increases computational performance and reduced model building time.

Evaluation Metrics:

	XGB Train	XGB Test
MSE	31.763895	67.994437
RMSE	5.635947	8.245874
MAE	2.776669	3.727759
R2	0.999656	0.999267
MAPR	0.005346	0.007202

```
{'objective': 'reg:squarederror', 'base_score': None, 'booster': None, 'colsample_bylevel': None, 'colsample_bynode': None, 'colsample_bytree': 0.75, 'eval_metric': None, 'gamma': 0.35, 'gpu_id': None, 'grow_policy': None, 'interaction_constraints': None, 'learning_rate': 0.06, 'max_bin': None, 'max_cat_threshold': None, 'max_cat_to_onehot': None, 'max_delta_step': None, 'max_depth': 50, 'max_leaves': None, 'min_child_weight': 2, 'monotone_constraints': None, 'n_jobs': None, 'num_parallel_tree': None, 'predictor': None, 'random_state': None, 'reg_alpha': 30, 'reg_lambda': 0.01, 'sampling_method': None, 'scale_pos_weight': None, 'subsample': 0.7, 'tree_method': 'gpu_hist', 'validate_parameters': None, 'verbosity': None}
```

7.6 Support-Vector-Machine (SVM)

In this section, we focus on building a regression model using Support Vector Machines (SVM). SVM is a powerful algorithm that is widely used for regression tasks due to its ability to handle both linear and non-linear relationships in the data.

The SVM model is constructed using a pipeline that incorporates a preprocessor and an SVR (Support Vector Regression) regressor. The preprocessor handles data preprocessing steps, such as feature scaling and encoding, to ensure compatibility with the SVM model.

To find the optimal hyperparameters for the SVM model, we utilize RandomizedSearchCV. This technique performs a randomized search over a specified parameter distribution, allowing us to explore different combinations of hyperparameters efficiently. The hyperparameters we tune include 'C', which controls the regularization strength, 'kernel' for the choice of kernel function, and 'epsilon' that sets the margin of error allowed in the model.

SVM/SVR is a theoretically simple algorithm, but its computational complexity makes it a time-consuming process, often taking several hours to build. The time complexity of SVM is typically in the range of $O(n^2)$ to $O(n^3)$, where n is the number of training samples.

The 'linear' kernel of SVR has could potentially outperfrom the 'rbf' and 'poly' kernels, however, due to its extreme computational inefficiency, we were not able to finish building a model this way. We then removed 'linear' from the kernel parameter list, which reduced computation times from +10 hours to a mere 8 minutes.

Additionally, SVR is sensitive to the introduced ranking of categorical variables through Ordinal encoding, which resulted in very poor performance.

As expected, the SVM Regression is very much effected by the ordinal encoding. The results differ a lot compared to the OneHot encoding. See [Encoding differences](#)

Evaluation Metrics:

	SVR Train	SVR Test
MSE	49434.884550	49742.882297
RMSE	222.339570	223.031124
MAE	131.292027	131.776217
R2	0.465137	0.463720
MAPR	0.206133	0.206692

7.7 AdaBoost Regressor

In the AdaBoost Regressor building section, we utilize the AdaBoost algorithm to train a regression model. AdaBoost stands for Adaptive Boosting and is a popular ensemble learning method that combines multiple weak learners to create a strong learner.

We start by defining a pipeline consisting of a preprocessor and the AdaBoostRegressor. The preprocessor handles the data preprocessing steps, such as feature scaling or encoding. The AdaBoostRegressor is the core component that performs the boosting algorithm.

We specify a parameter distribution that contains different hyperparameters for the AdaBoostRegressor, including the number of estimators, learning rate, loss function, base estimator (such as DecisionTreeRegressor or RandomForestRegressor), and random state.

Next, we perform randomized search cross-validation using RandomizedSearchCV to explore different combinations of hyperparameters.

Evaluation Metrics:

	AdaBoost Train	AdaBoost Test
MSE	32.013453	57.512628
RMSE	5.658043	7.583708
MAE	1.869100	2.461741
R2	0.999654	0.999380
MAPR	0.003630	0.004807

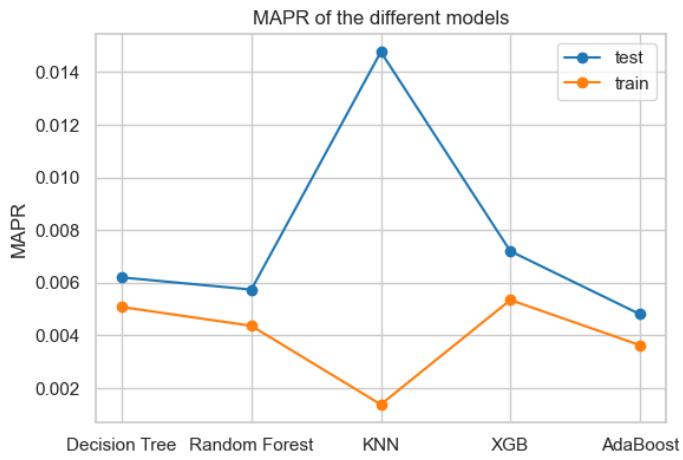
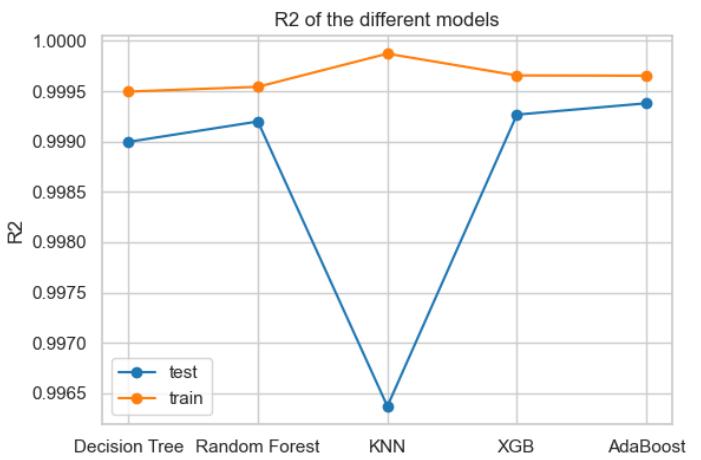
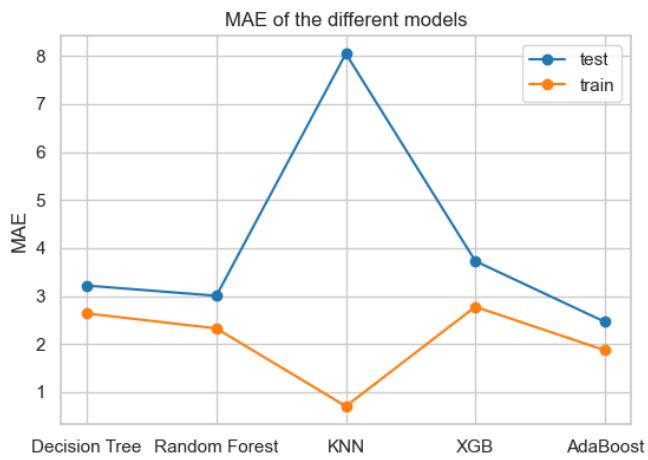
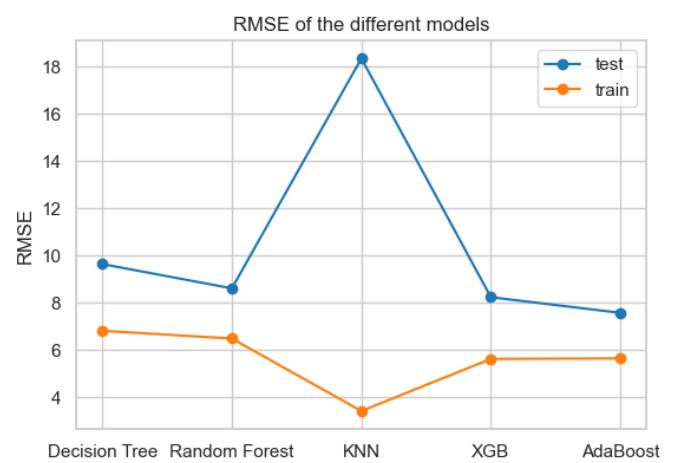
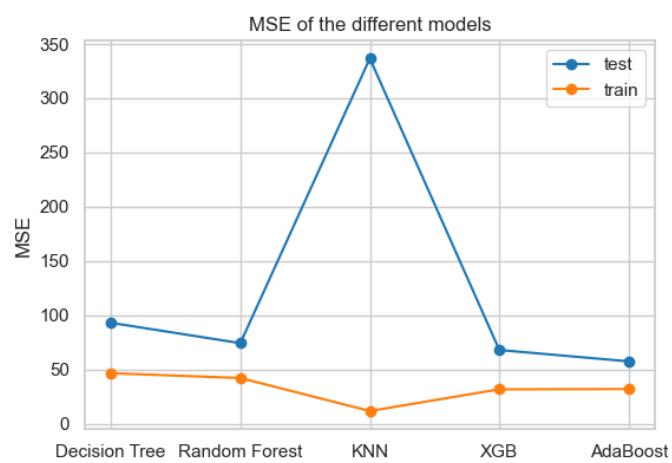
8 Test Data Performance

In this section, we compare the performances of our different machine learning models on the test data. Evaluating the models on unseen data is crucial to assess their generalization capabilities and determine their effectiveness in making predictions.

8.1 Metrics comparison

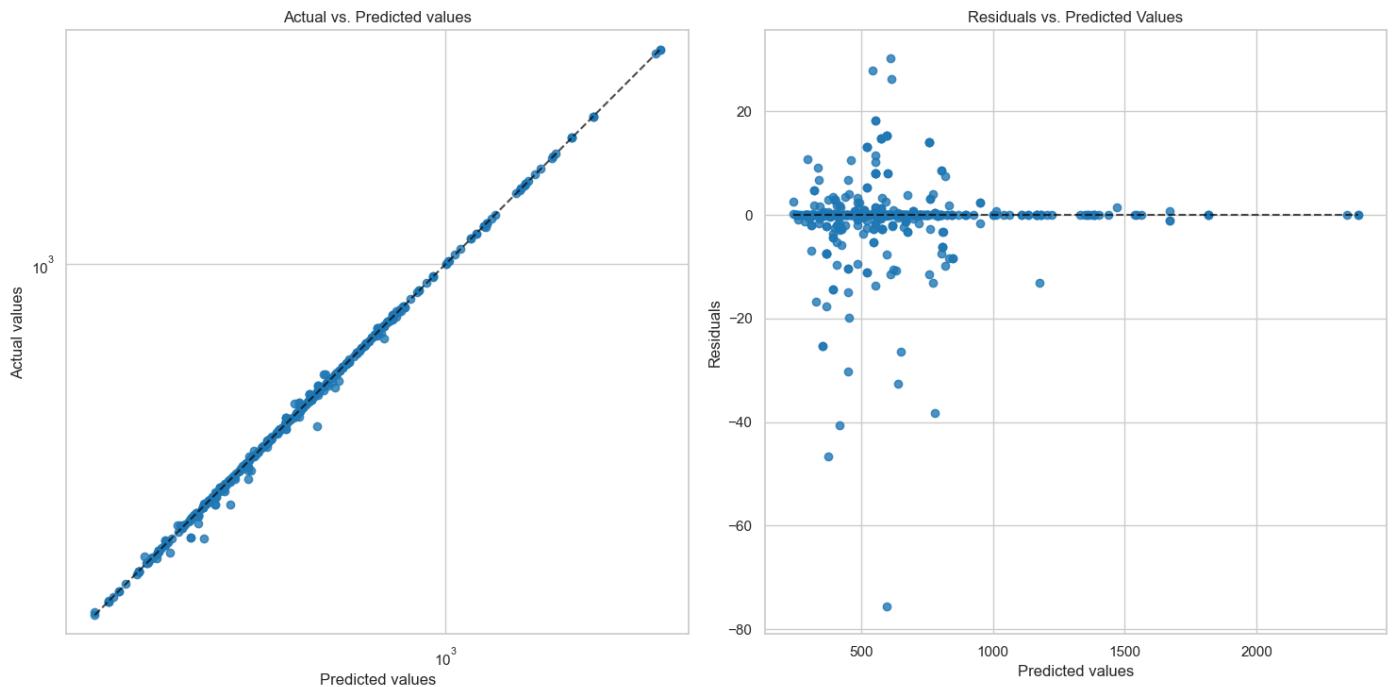
	Decision Tree Train	Random Forest Train	KNN Train	XGB Train	AdaBoost Train	SVR Train
MSE	46.588577	42.209474	11.761946	31.763895	32.013453	49434.884550
RMSE	6.825583	6.496882	3.429569	5.635947	5.658043	222.339570
MAE	2.636798	2.325318	0.702970	2.776669	1.869100	131.292027
R2	0.999496	0.999543	0.999873	0.999656	0.999654	0.465137
MAPR	0.005080	0.004366	0.001378	0.005346	0.003630	0.206133

	Decision Tree Test	Random Forest Test	KNN Test	XGB Test	AdaBoost Test	SVR Test
MSE	93.093921	74.309919	336.703067	67.994437	57.512628	49742.882297
RMSE	9.648519	8.620320	18.349470	8.245874	7.583708	223.031124
MAE	3.217996	3.005066	8.051065	3.727759	2.461741	131.776217
R2	0.998996	0.999199	0.996370	0.999267	0.999380	0.463720
MAPR	0.006203	0.005742	0.014772	0.007202	0.004807	0.206692



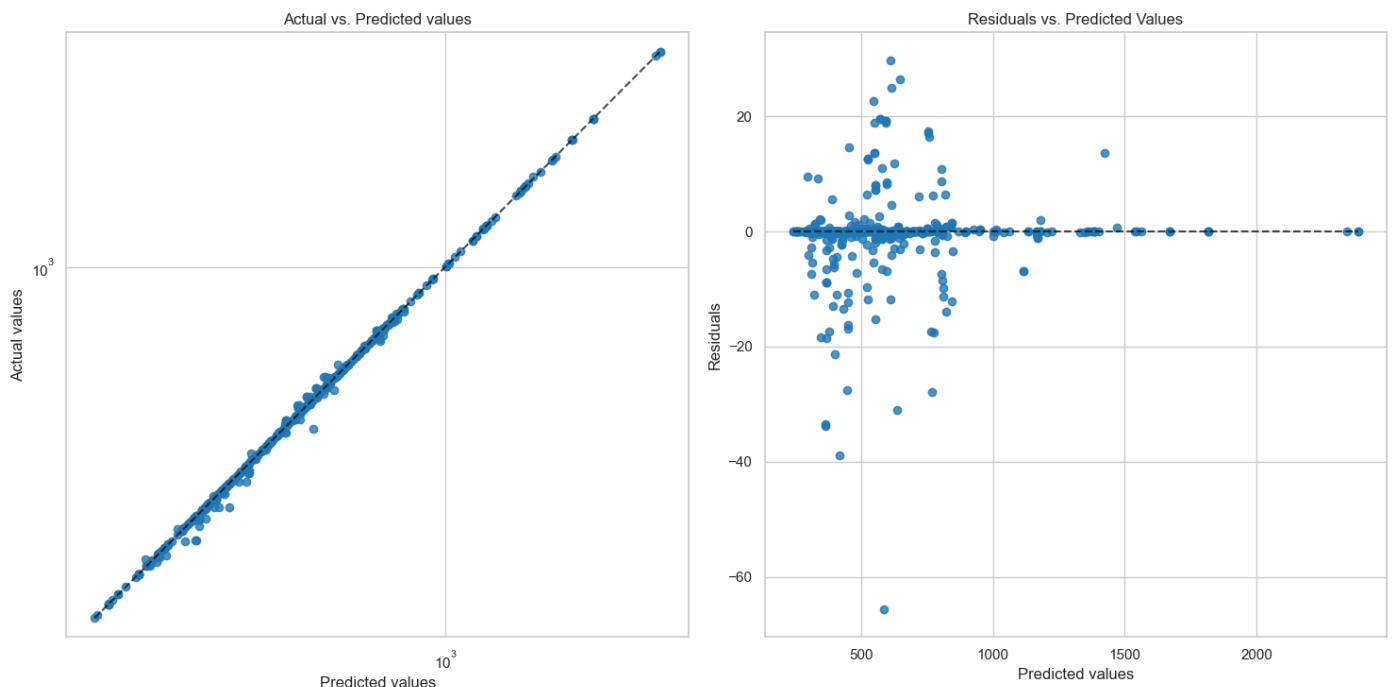
8.2 Predicted vs actual plots

Plotting cross-validated predictions of Decision Tree



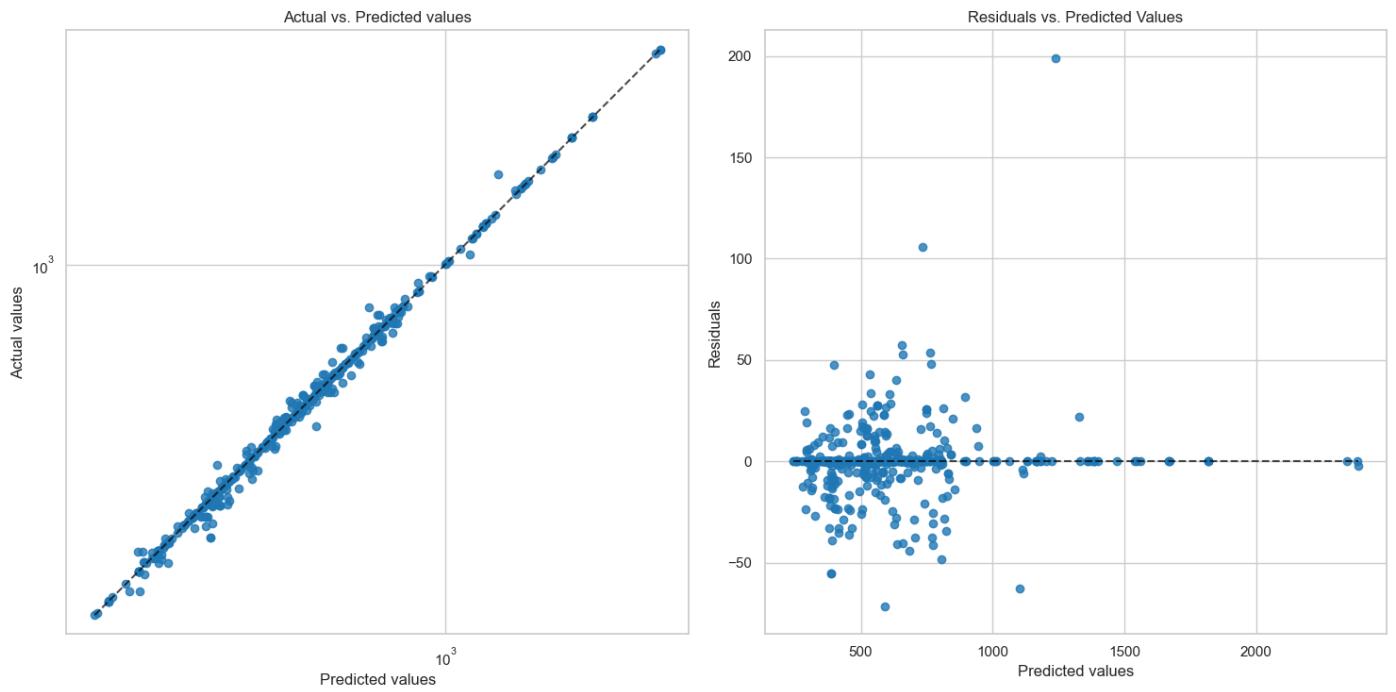
The mean residual of Decision Tree is: 0.27872454235489874

Plotting cross-validated predictions of Random Forest



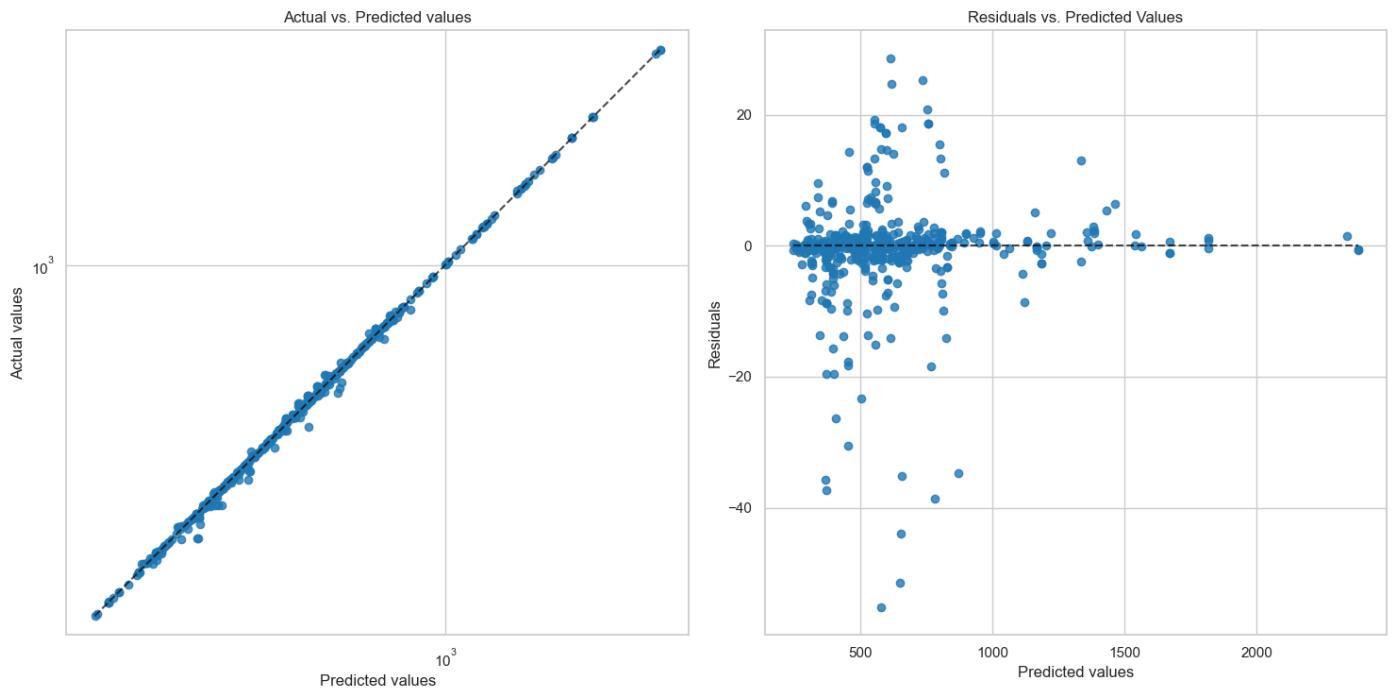
The mean residual of Random Forest is: 0.2560250310969117

Plotting cross-validated predictions of KNN



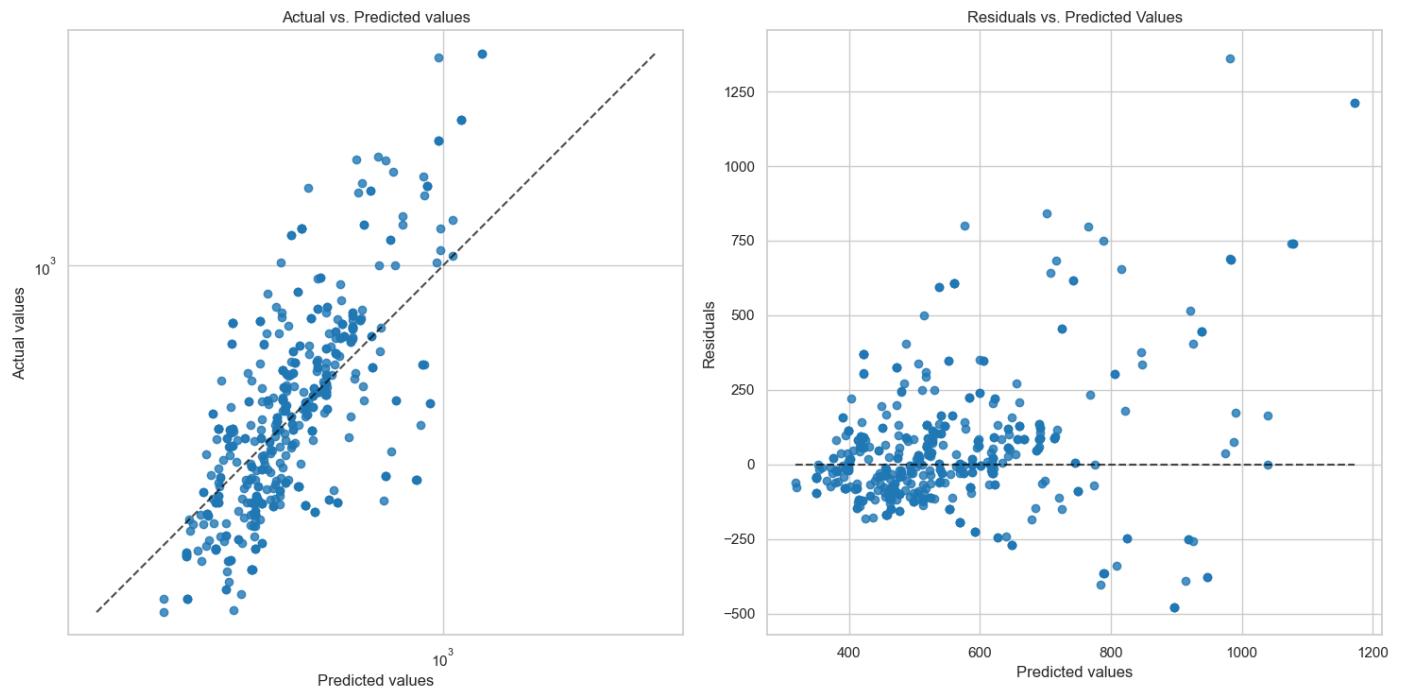
The mean residual of KNN is: 0.2874185173522171

Plotting cross-validated predictions of XGBoost



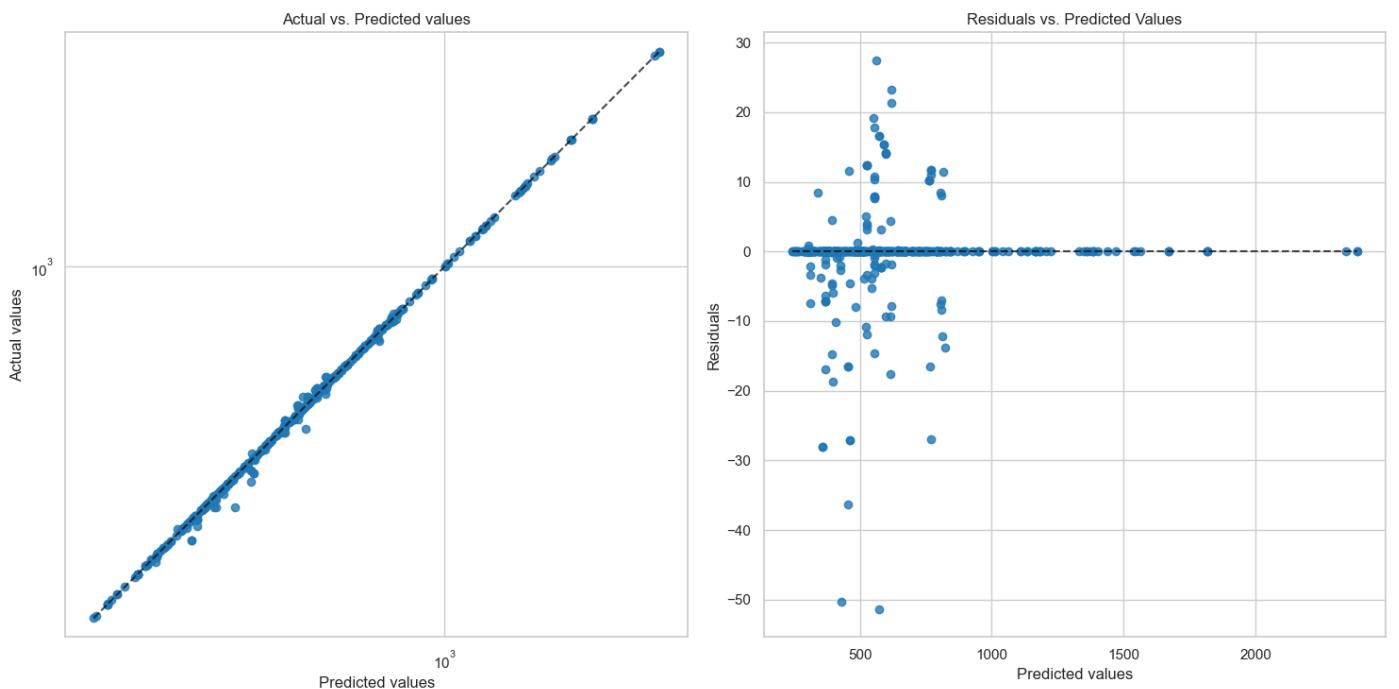
The mean residual of XGBoost is: 0.3873319158775656

Plotting cross-validated predictions of Support-Vector-Regressor



The mean residual of Support-Vector-Regressor is: 41.631615507053944

Plotting cross-validated predictions of Ada Boost Regressor



The mean residual of Ada Boost Regressor is: 0.1737747323158299

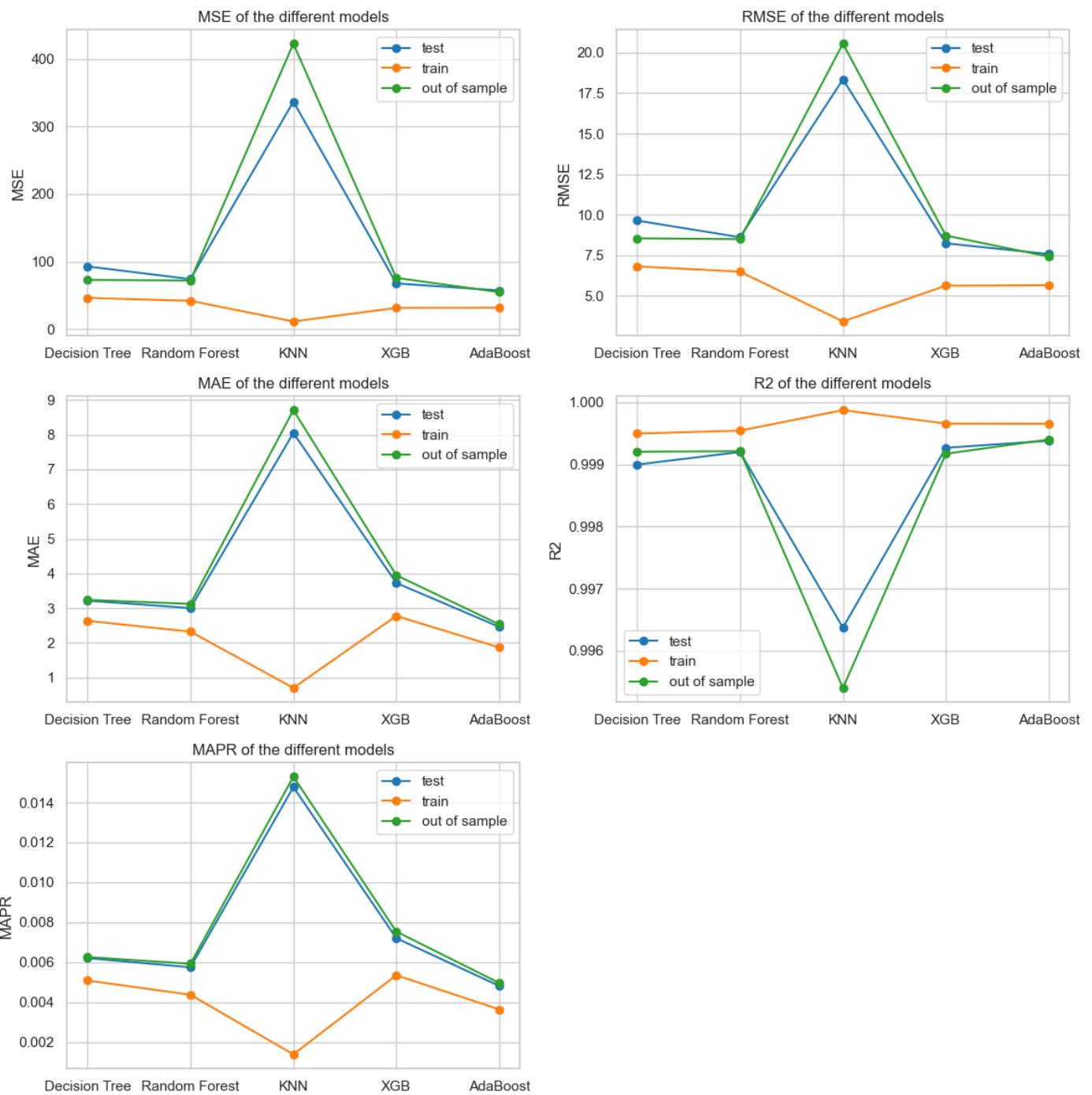
9 Out of sample performance

In this section, we analyze the performance of our machine learning models on out-of-sample data. Out-of-sample data refers to data that was not used during the model training and evaluation process. Evaluating the models on out-of-sample data provides a more realistic assessment of their performance and helps us understand how well they can generalize to new, unseen instances.

By examining the performance metrics on the out-of-sample data, we can determine how well our models are likely to perform in real-world scenarios. This evaluation allows us to validate the models' effectiveness, identify any potential issues or limitations, and make informed decisions about their deployment.

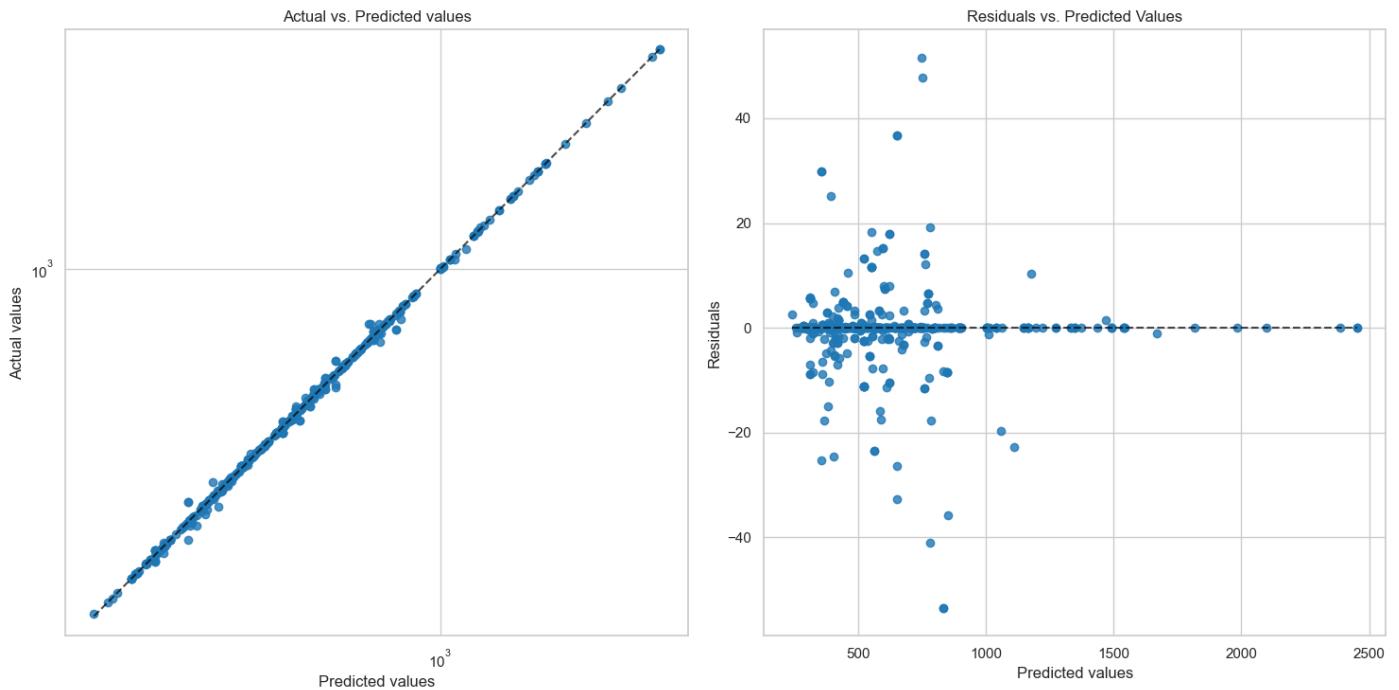
9.1 Metrics comparison

	Decision Tree out of sample	Random Forest out of sample	KNN out of sample	XGB out of sample	SVR out of sample	AdaBoost out of sample
MSE	73.118461	72.311337	422.391913	76.123017	49388.193690	55.101985
RMSE	8.550933	8.503607	20.552175	8.724851	222.234547	7.423071
MAE	3.240827	3.124188	8.708540	3.950068	131.664068	2.536617
R2	0.999204	0.999212	0.995400	0.999171	0.462136	0.999400
MAPR	0.006254	0.005918	0.015295	0.007532	0.207219	0.004957



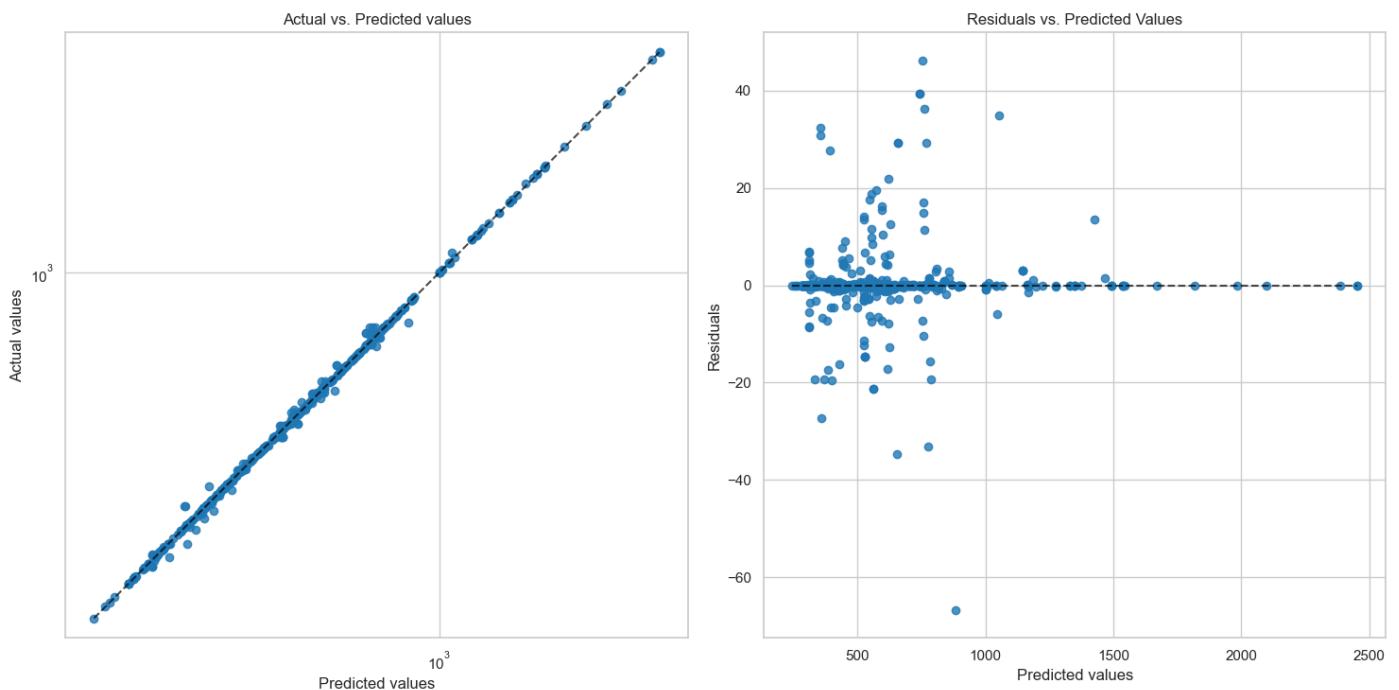
9.2 Predicted vs actual plots

Plotting cross-validated predictions of Decision Tree



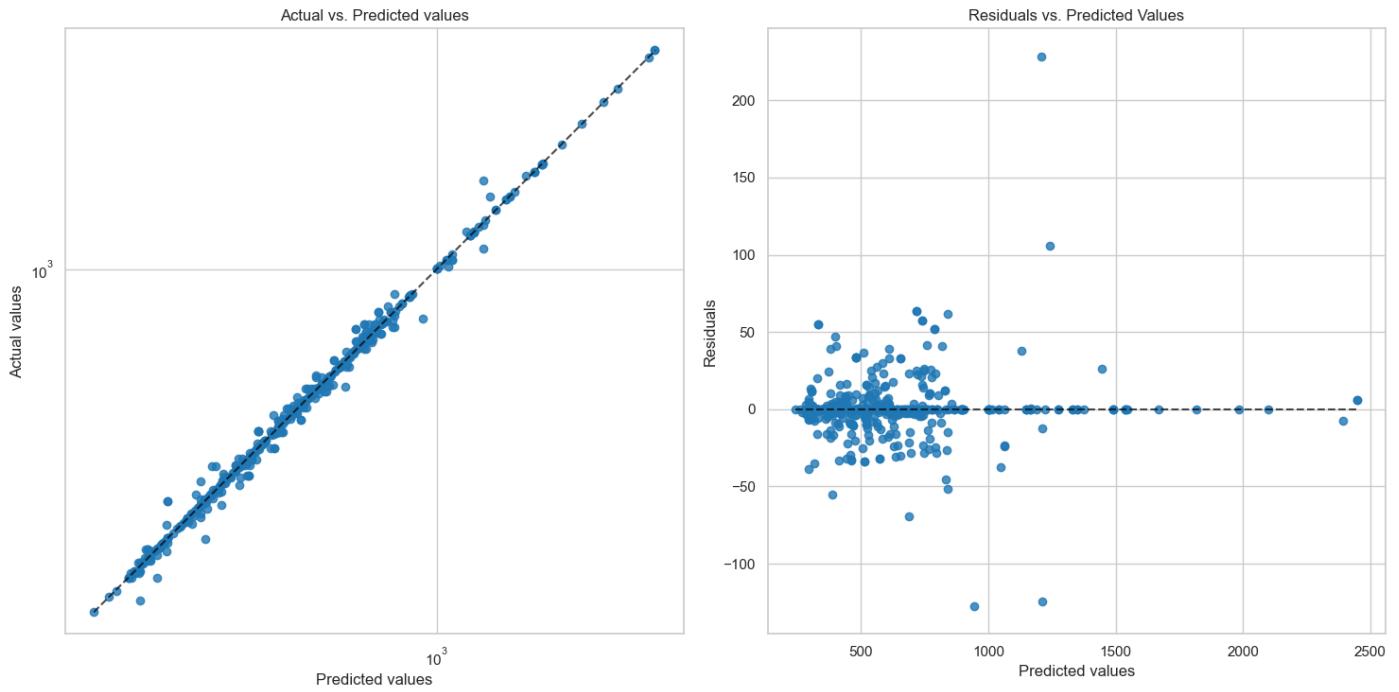
The mean residual of Decision Tree is: -0.22234999227525504

Plotting cross-validated predictions of Random Forest



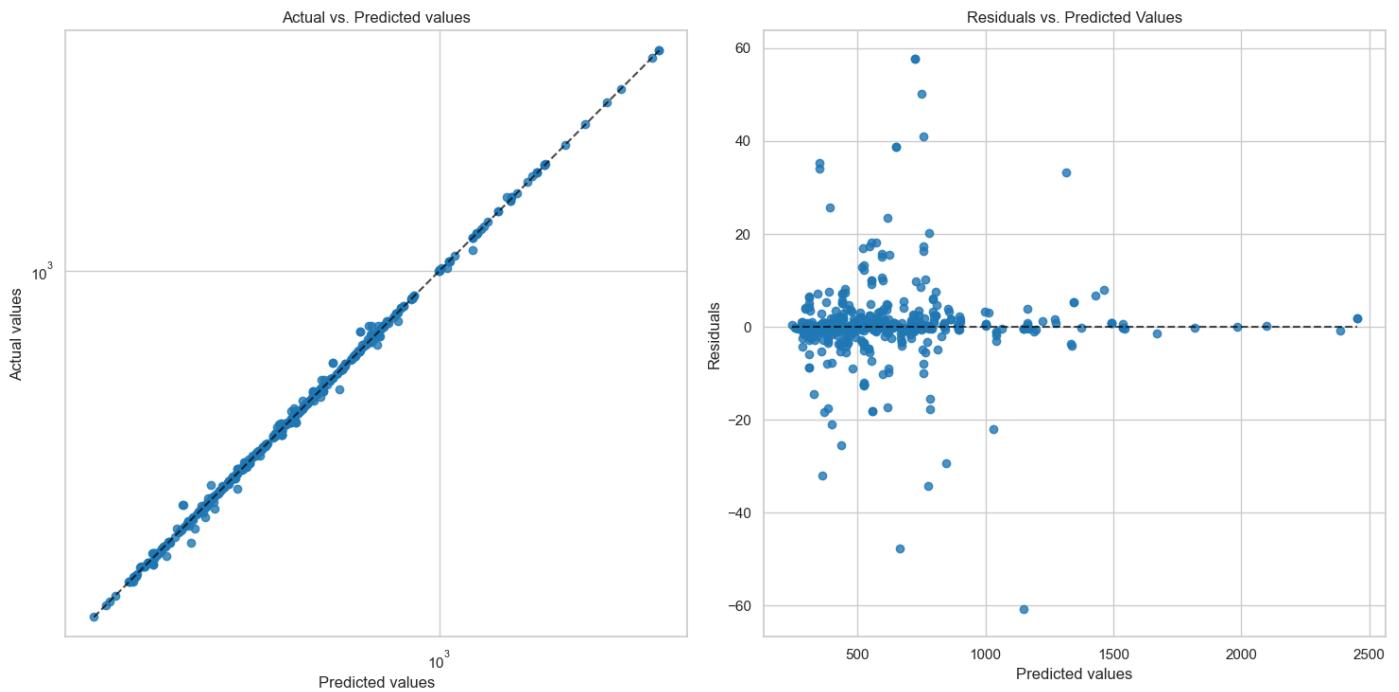
The mean residual of Random Forest is: -0.12240398551644305

Plotting cross-validated predictions of KNN



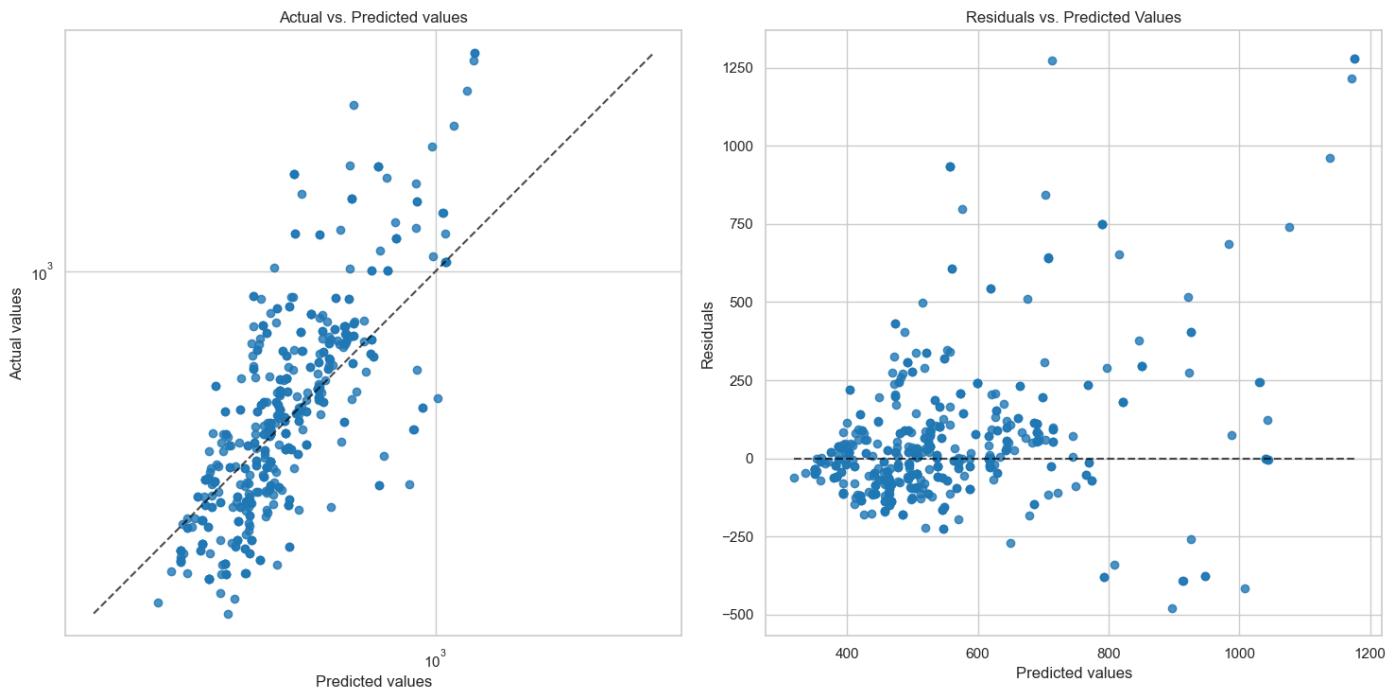
The mean residual of KNN is: -0.1064419259358378

Plotting cross-validated predictions of XGBoost



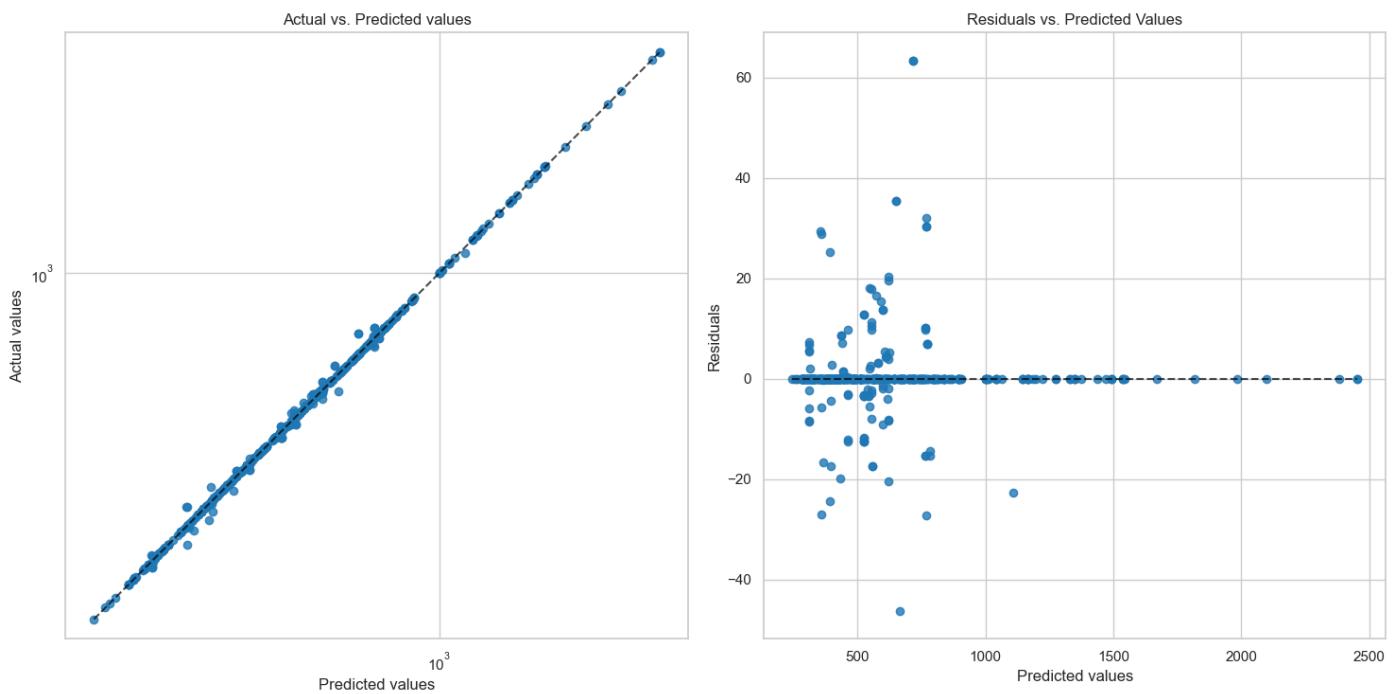
The mean residual of XGBoost is: -0.04810206074006601

Plotting cross-validated predictions of Support-Vector-Regressor



The mean residual of Support-Vector-Regressor is: 40.993052932640516

Plotting cross-validated predictions of Ada Boost Regressor



The mean residual of Ada Boost Regressor is: -0.1042347929474167

10 Feature Importance Analysis

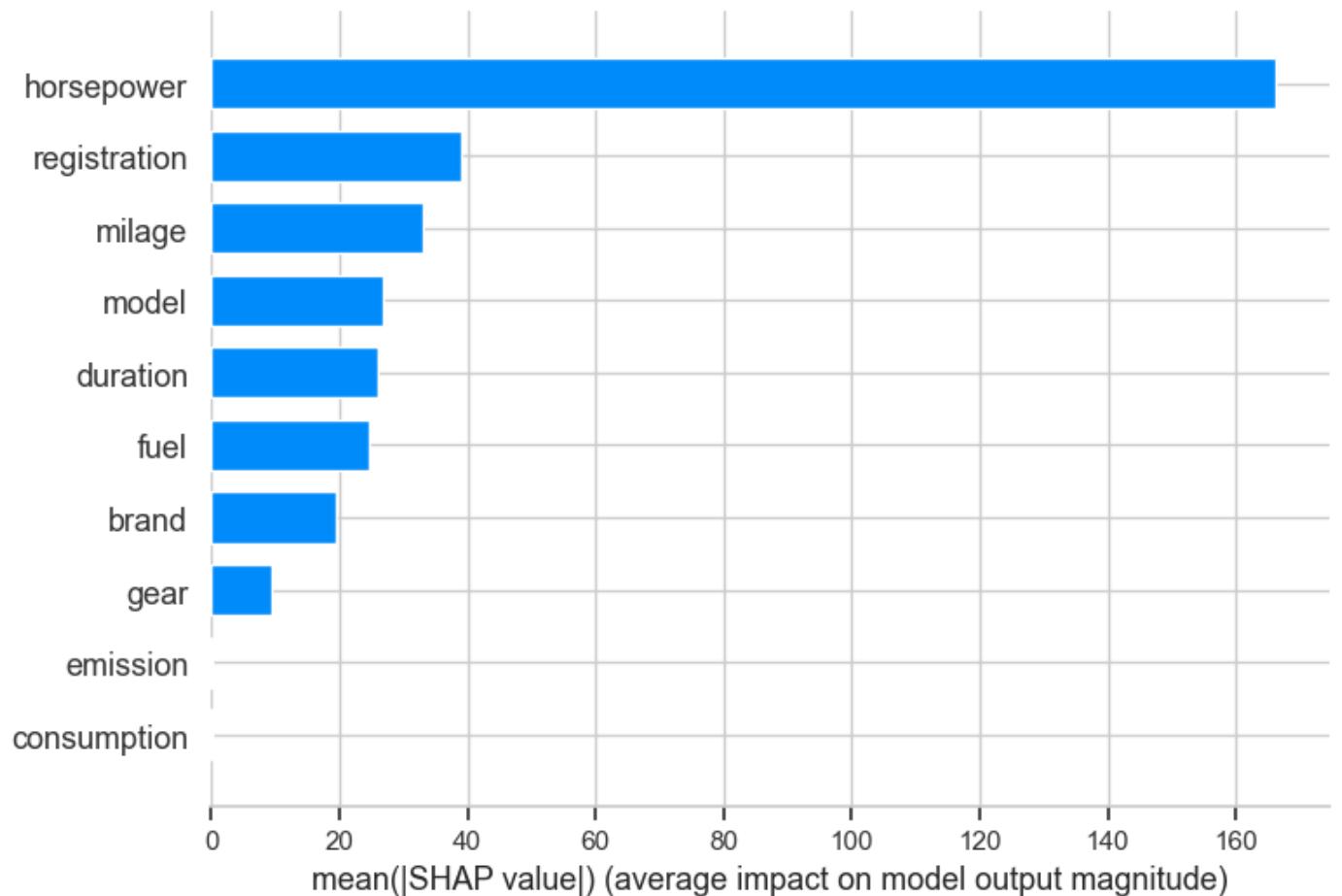
In this section, we delve into evaluating the importance of features in our machine learning models. Understanding the significance of different features can provide valuable insights into their impact on the prediction outcome. We use two methods for feature importance assessment: the SHAP (SHapley Additive exPlanations) library and the built-in feature importance functions.

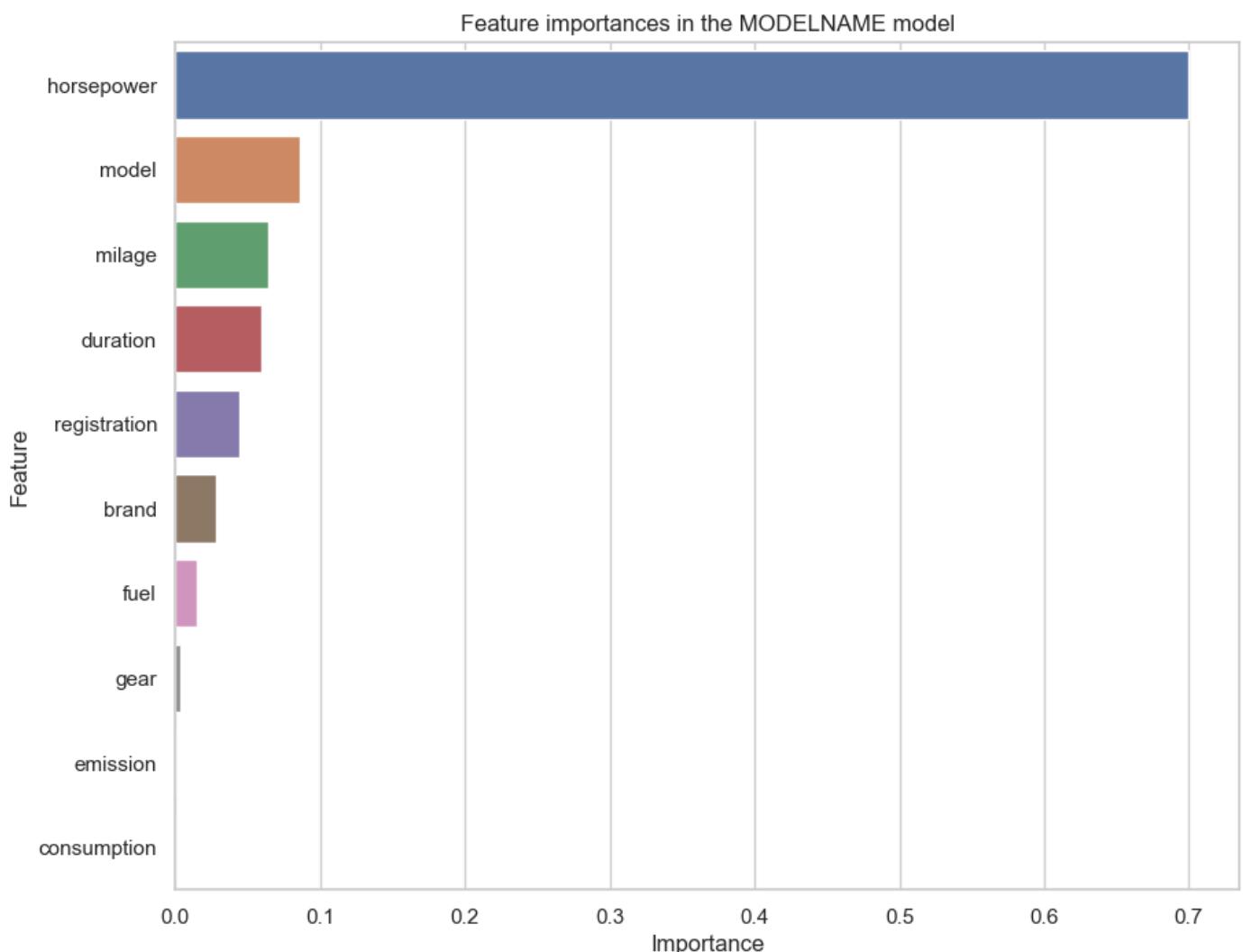
The SHAP library offers a powerful tool for explaining individual predictions by quantifying the contribution of each feature. It provides a comprehensive view of feature importance by considering all possible feature combinations and their respective contributions. Additionally, we utilize the built-in feature importance functions provided by the selected machine learning models. These functions calculate the relevance of features based on various metrics specific to each algorithm.

Helper functions

10.1 Decision tree feature importance

```
[[ 40.04565061  55.18708265  149.26930324 ...   -8.33394578
    4.47276674 -22.30324541]
 [ 37.26114636 -2.93153318 -19.09585675 ...  -22.61119512
  14.44855256 -35.59556054]
[-273.57624226 -43.19609511  135.21828697 ...  -52.98984253
 -5.51464869 -9.93819321]
...
[-159.79132737 -39.57681445 -12.71274715 ...  -45.10263063
  3.77469752  17.16793339]
[-128.47139901 -37.47258166 -14.1889055 ...  -19.42747158
 20.42734471  60.60715812]
[ 28.66515735 -43.8049596  -8.8668519 ...   35.5877998
 -21.92577387  79.0579165 ]]
```





We have two interpretations of feature importance from a Decision Tree model: one is based on Mean SHapley Additive exPlanations (SHAP) values, and the other is based on the inbuilt feature importance of the model. Both interpretations reveal insights into how features contribute to the model's predictive performance.

'Horsepower' is deemed as the most influential feature in both interpretations. With a SHAP value of over 160 and a feature importance score of about 0.7, it is clear that changes in 'Horsepower' significantly impact the model's predictions. Therefore, 'Horsepower' is a crucial feature for the decision-making process of this model.

The 'Registration' and 'Model' features are identified as the second most important variables, but in different interpretations. 'Registration' has a significant impact according to SHAP values, while 'Model' stands out in the inbuilt feature importance measure. This disparity may be due to the different ways these metrics calculate importance.

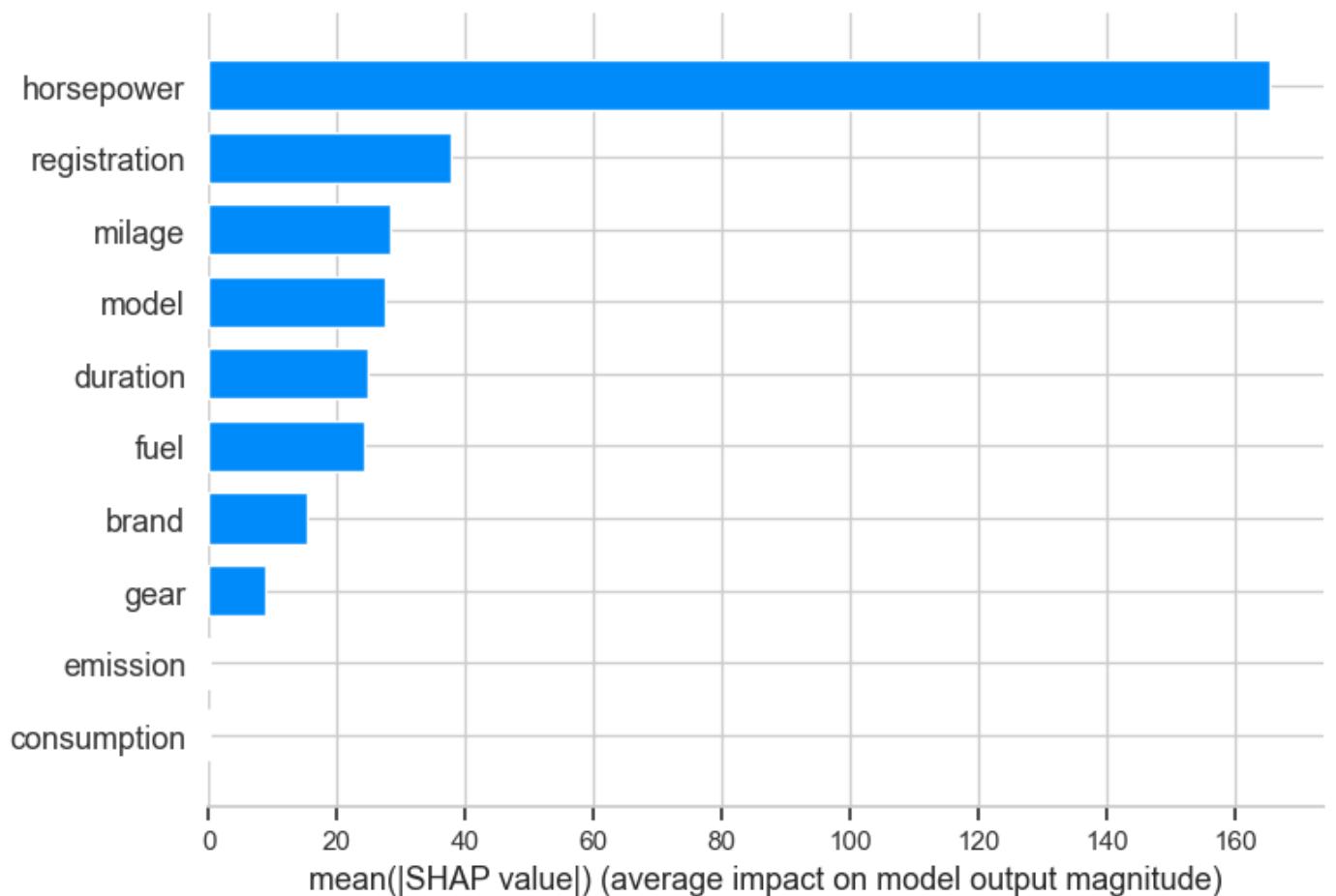
'Mileage' and 'Duration' both have comparable importance levels according to SHAP values and the inbuilt feature importance, with values around 30 and 0.06 respectively. This consistency suggests that while these features play a role in the model's decisions, their impact is less substantial compared to 'Horsepower' and 'Registration' or 'Model'.

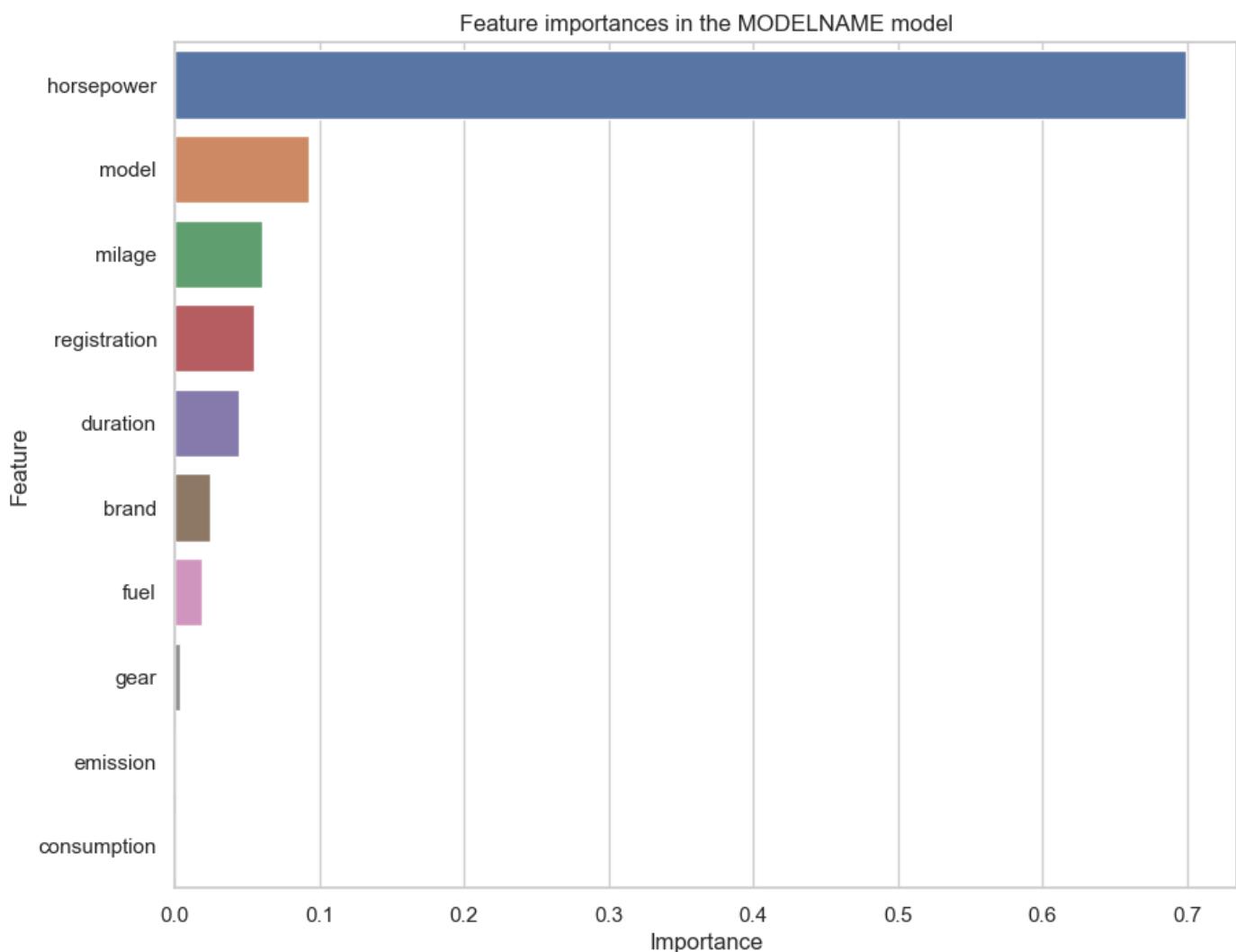
Lastly, 'Emission' and 'Consumption' have been identified as having negligible influence in both interpretations. Their low SHAP values and feature importance scores suggest that these features contribute minimally to the model's predictive ability.

In summary, 'Horsepower' is the key feature in this model, followed by 'Registration' or 'Model', and then 'Mileage' and 'Duration'. The features 'Emission' and 'Consumption' have little to no impact on the model's decision-making process, indicating potential for simplifying the model without significantly impacting its accuracy. These interpretations can guide feature selection and engineering in future model iterations, and remind us that different feature importance methods may yield different perspectives.

10.2 Random forest feature importance

```
[[ 30.20481206  54.13501153 128.31131357 ... -10.464822  
  7.64512841 -23.69713009]  
 [ 29.21892628 -3.82952005 -19.51206999 ... -28.58248765  
  12.70812207 -29.58225115]  
 [-226.51821513 -42.52402692 100.24165516 ... -48.66544275  
  -6.41243689 -15.57807162]  
 ...  
 [-134.48636804 -60.49224673 -13.32978185 ... -30.42419318  
  4.20801625 13.32093094]  
 [-118.73185478 -24.1593265 -14.37778 ... -25.14262134  
  15.55067059 48.78875152]  
 [ 29.68835419 -24.69339949 -8.36051691 ... 27.31659195  
  -18.02434253 54.47163009]]
```





From both the Mean SHAP values and the inbuilt feature importance of the Random Forest model, we observe similar patterns:

'Horsepower' is the most influential feature, with high SHAP values (~165) and importance score (~0.7), making it crucial for the model's decision-making.

'Registration' and 'Model' are secondary in importance. The SHAP values highlight 'Registration' more (~40), while the inbuilt importance emphasizes 'Model' more (~0.09).

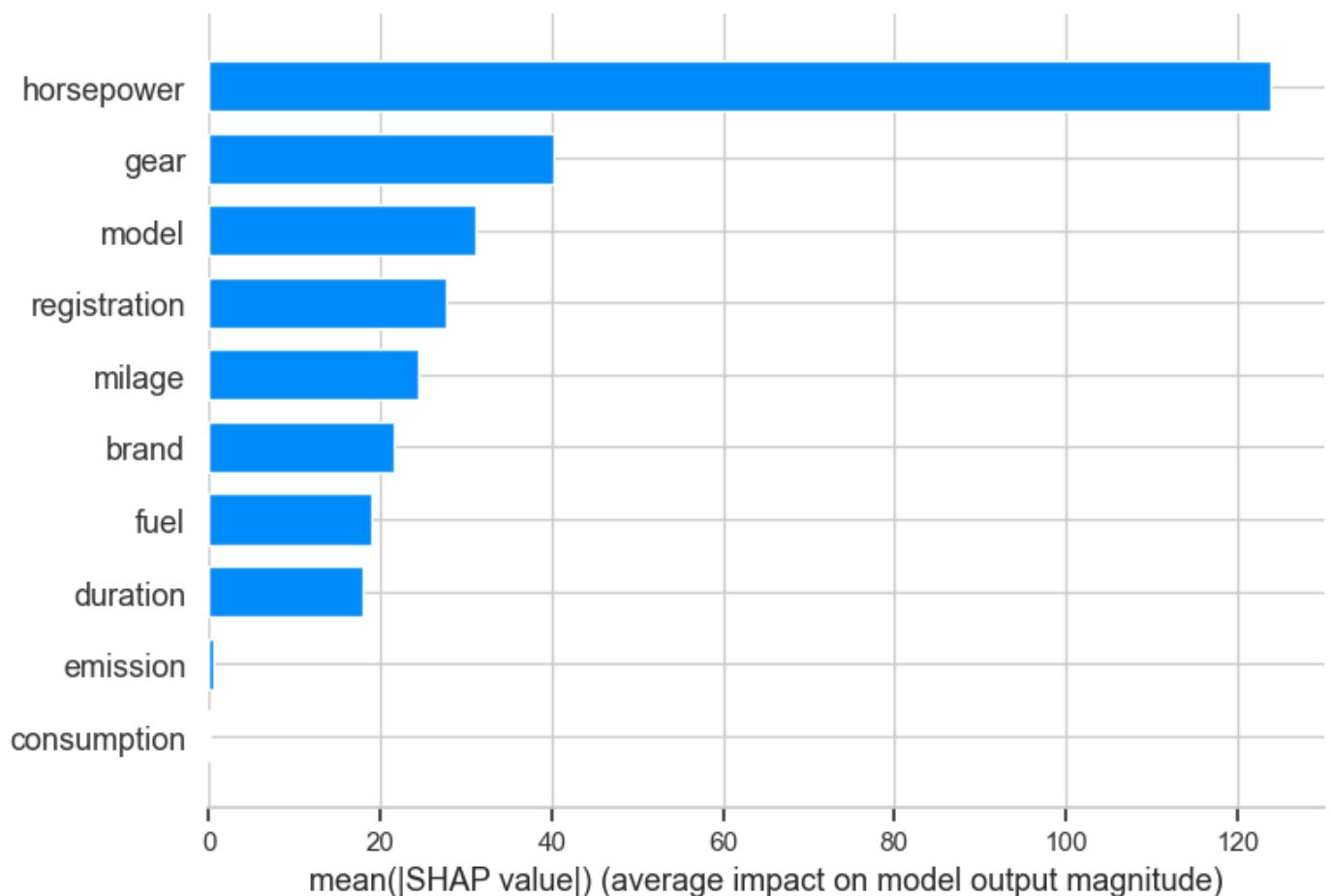
'Mileage' is similarly impactful in both measures (~35 SHAP, ~0.06 importance), indicating its moderate contribution.

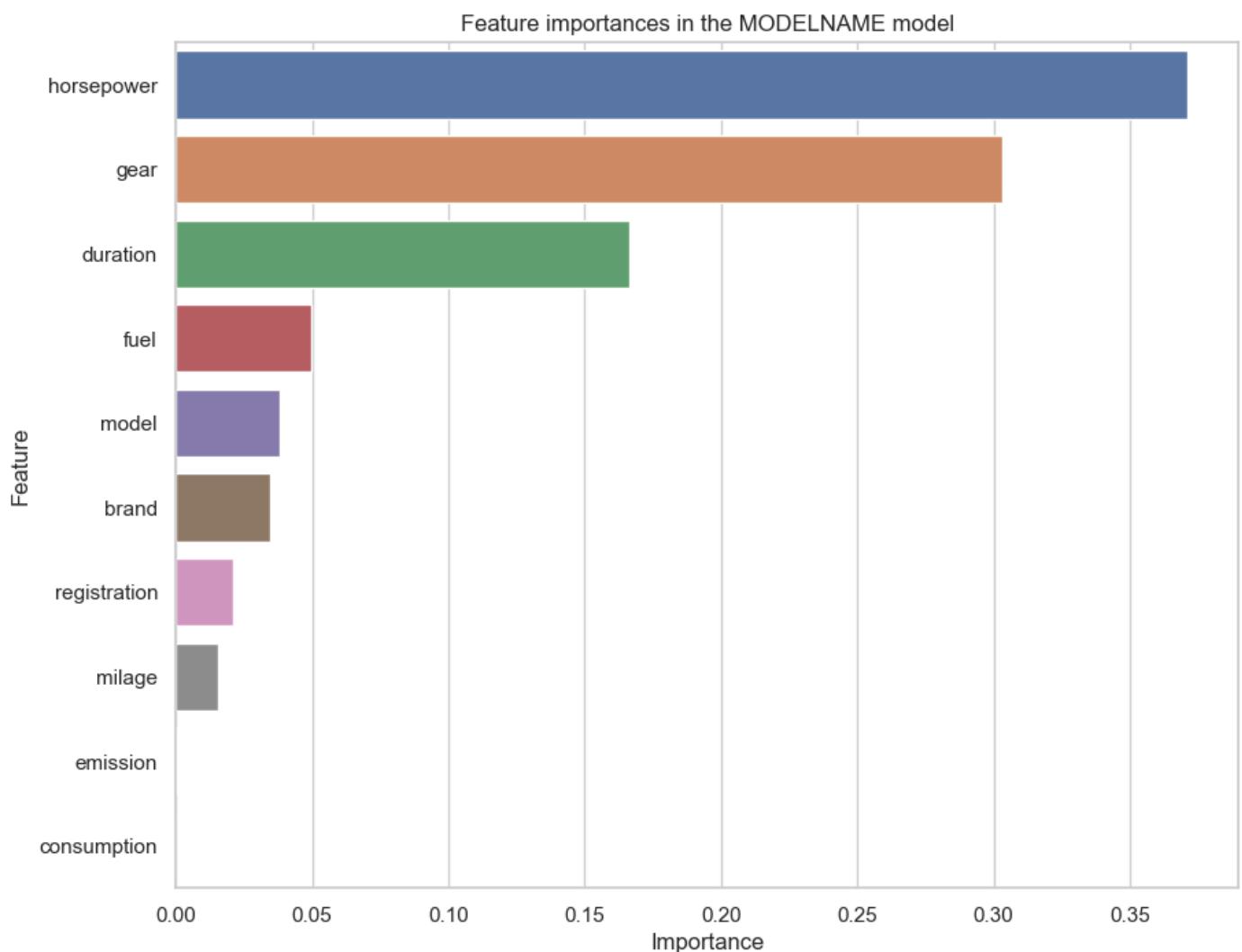
Lastly, 'Emission' and 'Consumption' are negligible in both interpretations, indicating their minimal impact on the model's predictive ability.

10.3 XGB feature importance

`ntrree_limit is deprecated, use `iteration_range` or model slicing instead.`

```
[[ 20.693304    6.7814074   91.90773   ... -49.232983   34.611034  
-18.459797 ]]  
[ 20.671803   -4.079041   -10.893994  ... -32.795994   37.805534  
-23.554195 ]  
[-100.386604  -65.988075   57.003014   ... -44.09324   -38.315025  
-15.491648 ]  
...  
[ -94.71441   -63.58814   -33.7905   ... -15.941605   29.690456  
-3.7415783 ]  
[ -66.30939   -31.984943  -12.446889  ... -48.805115   28.922964  
17.032265 ]  
[ 22.20005   -16.18955   -6.1919723  ...  25.281693   -34.74461  
31.632162 ]]
```





The interpretations for both Mean SHAP values and the built-in feature importance for the XGBoost model are as follows:

In the SHAP interpretation, 'Horsepower' is the most significant feature (~120), followed by 'Gear' (~40), 'Model' (~30), and 'Registration' (~25). 'Emission' and 'Consumption' are not significant.

The built-in feature importance of XGBoost, often determined by F-score (a measure of how frequently each feature appears in the model splits), indicates a similar importance of 'Horsepower' (~0.35) and 'Gear' (~0.3), followed by 'Duration' (~0.16). Again, 'Emission' and 'Consumption' aren't significant.

So, in both interpretations, 'Horsepower' is paramount, 'Gear' is important, while 'Emission' and 'Consumption' have little influence. The SHAP values emphasize the 'Model' and 'Registration' features, while the built-in importance underscores 'Duration'.

The method to compute this feature importance is through an F-score, which essentially measures how frequently each feature appears in the models created during the boosting process.

In XGBoost, each decision tree is built by repeatedly splitting the data into two groups. Each split involves a single feature at a time. The more frequently a feature is used in making splits across all trees, the higher its F-score, and thus the more important it is considered to be. This is because a feature that is often used for splitting is one that does a good job of separating the data, thereby improving the model's performance.

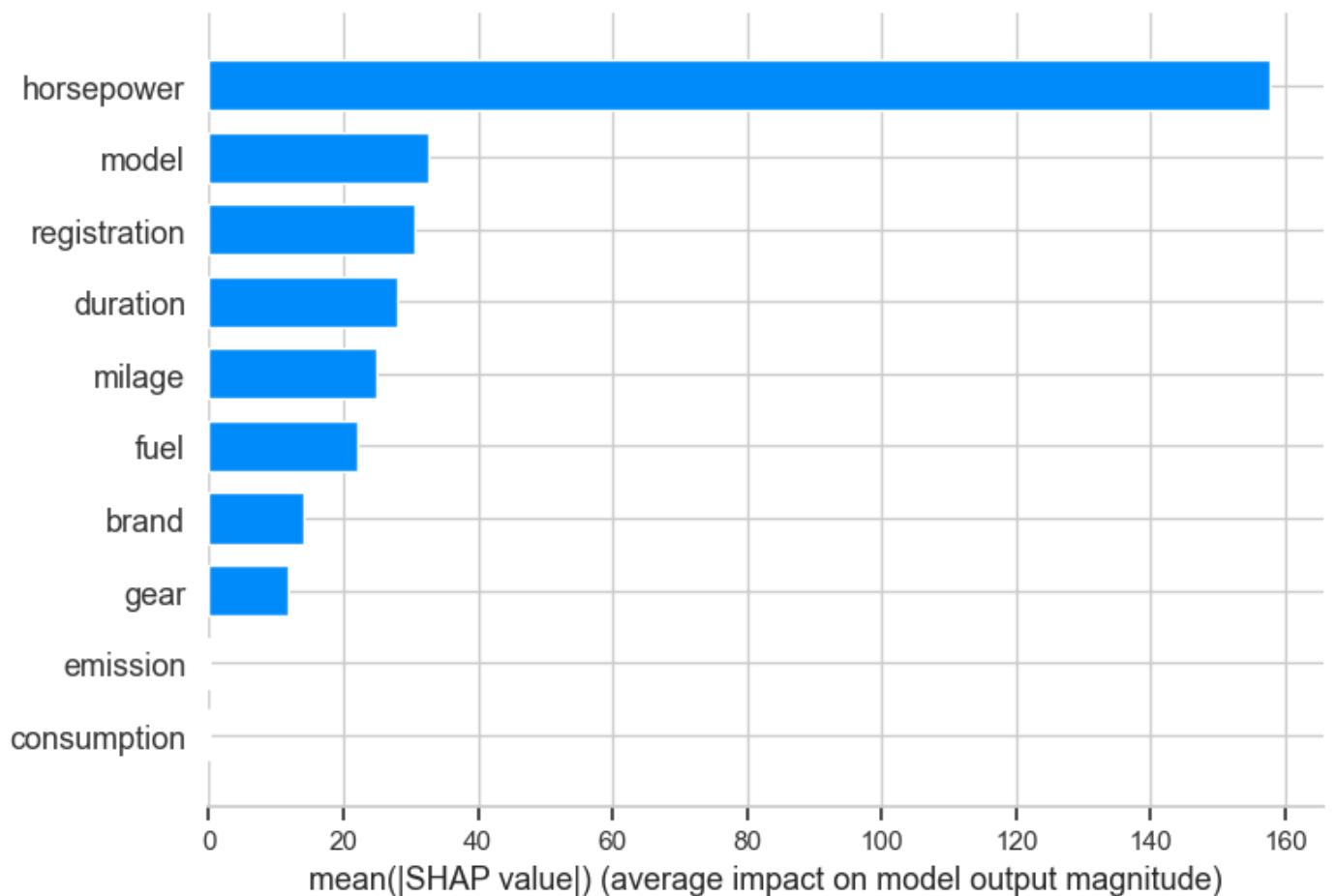
In your XGBoost model, 'Horsepower' has the highest built-in feature importance, followed by 'Gear' and then 'Duration'. This means that these three features are the ones most often used to split the data in your model, and thus they have the most significant impact on your model's predictions. Conversely, 'Emission' and 'Consumption' are not important, meaning they are rarely used in data splits and have little effect on the predictions.

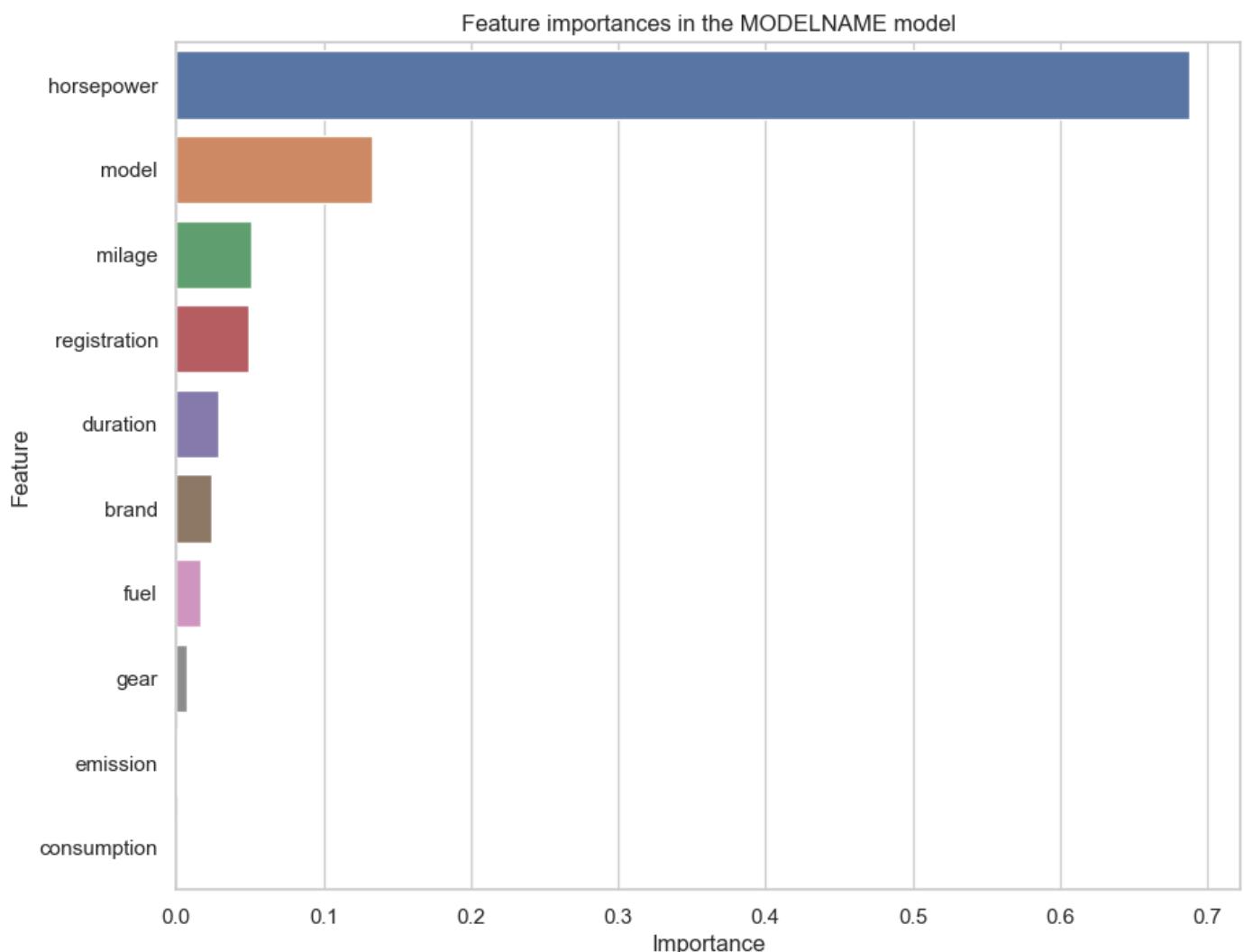
It's worth noting that while built-in feature importance gives us a good indication of which features are most useful for making predictions, it doesn't tell us anything about the nature of the relationships between these features and the target variable.

10.4 AdaBoost feature importance

100% |██████████| 2856/2856 [43:21<00:00, 1.10it/s]

```
[ [ 15.78760574  23.88782611  64.99084034 ... -11.73075167  
  3.16750577 -32.00647549]  
 [ 19.25240778 -10.94560616 -18.57310809 ... -42.68079854  
  19.55885616 -18.86740759]  
 [-177.22440445 -33.47853882  54.68803795 ... -47.34926821  
 -11.51212315 -15.99079554]  
 ...  
 [-142.40686064 -78.41915357 -20.52064346 ... -8.20000184  
  3.9158641  11.77362094]  
 [-135.53479573 -26.39817339 -19.12514773 ... -26.43670631  
  9.12260014  46.17229391]  
 [ 19.33702382 -16.65641411 -16.71414699 ... 18.87225968  
 -53.46724772  44.82203532] ]
```





The interpretations for both Mean SHAP values and the built-in feature importance for the AdaBoost model are as follows:

SHAP values indicate 'Horsepower' as the most significant feature (~160), followed by 'Model' (just below 40), 'Registration' (closely behind 'Model'), and 'Mileage' (~35). 'Emission' and 'Consumption' don't hold significant importance.

The built-in feature importance of AdaBoost, computed based on the weight of the evidence each feature provides across all the decision stumps, shows 'Horsepower' with the highest importance (just below 0.7). 'Model' follows (just below 0.15), and then 'Mileage' and 'Registration' (both ~0.05). Again, 'Emission' and 'Consumption' aren't significant.

The built-in feature importance in AdaBoost is computed based on the contribution of each feature to the weighted error rate of the model. In AdaBoost, each feature is used as a decision stump, and an importance score is calculated for each feature based on how much it decreases the weighted error of the model. The more a feature decreases this weighted error, the more important it is considered to be.

In conclusion, 'Horsepower', 'Model', 'Registration', and 'Mileage' are key features in the AdaBoost model according to both SHAP and built-in feature importance, with 'Emission' and 'Consumption' providing little influence. However, the SHAP and built-in feature importance differ slightly in the relative importance they assign to 'Model', 'Registration', and 'Mileage'.

11 Safe models

12 Debugging library versions

Sklearn Version has to be >1.2.2 For that, Python>3.8 is required

Pandas

```
Requirement already satisfied: pandas==2.0.1 in c:\users\tobia\anaconda3\envs\py39
\lib\site-packages (2.0.1)
Requirement already satisfied: numpy>=1.20.3 in c:\users\tobia\anaconda3\envs\py39
\lib\site-packages (from pandas==2.0.1) (1.23.5)
Requirement already satisfied: tzdata>=2022.1 in c:\users\tobia\anaconda3\envs\py39
\lib\site-packages (from pandas==2.0.1) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\tobia\anaconda3\envs\py39\lib\site-packages (from pandas==2.0.1) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\tobia\anaconda3\envs\py39\lib\site-packages (from pandas==2.0.1) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\tobia\anaconda3\envs\py39\lib\site-packages (from python-dateutil>=2.8.2->pandas==2.0.1) (1.16.0)
```

Sklearn

Numpy

Shap

Numba, llvmlite (used in shap)

False

XGBoost

Helper libraries

References

This work draws inspiration from the master thesis conducted by Thomas Dornigg. To delve deeper into Dornigg's thesis, please refer to the following link: [Link to Thomas Dornigg's Master Thesis](#)

- Train/test split: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- OneHot and ordinal encoding: <https://stackoverflow.com/questions/69052776/ordinal-encoding-or-one-hot-encoding>
- OneHot vs ordinal encoding: <https://github.com/slundberg/shap/issues/397>

- XGBoost hyperparameter tuning: <https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning>
- get hyperparameters XGBoost: <https://stackoverflow.com/questions/69639901/retrieve-hyperparameters-from-a-fitted-xgboost-model-object>
- SHAP explained: <https://shap.readthedocs.io/en/latest/index.html>
- Sklearn documentation: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- Why label encoding should not be used on input data:
<https://stackoverflow.com/questions/59914210/why-shouldnt-the-sklearn-labelencoder-be-used-to-encode-input-data>
- Ordinal encoder: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html>
- How shap values would work with OneHot encoding:
https://www.reddit.com/r/datascience/comments/s2epy0/computing_categorical_feature_import/
- shap values of categorical variables: <https://github.com/slundberg/shap/issues/397>
- Looked into parallelizing shap calculations: <https://towardsdatascience.com/parallelize-your-massive-shap-computations-with-mllib-and-pyspark-b00accc8667c>
- Sklearn Decision Tree Regressor: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>
- Sklearn AdaBoost: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html>
- Sklearn Random Forest: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>
- Sklearn KNN Regressor: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html>
- Sklearn SVR: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>
- XGBoost CUDA GPU acceleration: <https://xgboost.readthedocs.io/en/stable/gpu/index.html>
- XGBoost: <https://xgboost.readthedocs.io/en/stable/index.html>
- Sklearn Decision Tree Regression explained: https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html

- Support Vector Machines explained: https://en.wikipedia.org/wiki/Support_vector_machine
- XGBoost feature importance: <https://mljar.com/blog/feature-importance-xgboost/#:~:text=About%20Xgboost%20Built%20In%20Feature%20Importance&text=You%20can%20use%20the%20shap%20library%20to%20interpret%20the%20importance%20of%20each%20feature%20in%20the%20model%20using%20shap%20explanations%20and%20plots%20such%20as%20the%20shap%20force%20plot%20or%20the%20shap%20summary%20plot%20to%20understand%20the%20model%20and%20its%20decisions%20better%20than%20ever%20before.>
- Feature importance explained with shap: <https://www.aidancooper.co.uk/a-non-technical-guide-to-interpreting-shap-analyses/>

Appendix

A1 Encoding differences

As we progressed in the creation of this machine learning notebook, we shifted from OneHot encoding to ordinal encoding. Notably, while some algorithms, like tree-based models, demonstrate adaptability to the choice of encoding, others show heightened sensitivity. In particular, Support Vector Machine (SVM) and Support Vector Regression (SVR) algorithms can be influenced by the biases initiated by varying encoding techniques. This sensitivity emanates from the dependency of SVM and SVR on vector geometry, where actual geometric distances between data points significantly impact their computational process.

OneHot encoding is used.

Decision Tree

Evaluation Metrics:

	Decision Tree Train	OneHot	Decision Tree Test	OneHot
MSE	48.476505		93.750237	
RMSE	6.962507		9.682471	
MAE	2.607555		3.232638	
R2	0.999476		0.998989	
MAPR	0.005063		0.006260	

```
{'ccp_alpha': 0.05648616489184735, 'criterion': 'squared_error', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 2, 'min_samples_split': 20, 'min_weight_fraction_leaf': 0.0, 'random_state': 2023, 'splitter': 'best'}
```

Random Forest

Evaluation Metrics:

	Random Forest Train OneHot	Random Forest Test OneHot
MSE	69.837894	103.283831
RMSE	8.356907	10.162865
MAE	2.975286	3.694924
R2	0.999244	0.998886
MAPR	0.005342	0.006765

```
{"bootstrap": True, "ccp_alpha": 0.0, "criterion": "squared_error", "max_depth": 40, "max_features": 1.0, "max_leaf_nodes": None, "max_samples": None, "min_impurity_decrease": 0.0, "min_samples_leaf": 4, "min_samples_split": 10, "min_weight_fraction_leaf": 0.0, "n_estimators": 700, "n_jobs": None, "oob_score": False, "random_state": 2023, "verbose": 0, "warm_start": False}
```

KNN**Evaluation Metrics:**

	KNN Train OneHot	KNN Test OneHot
MSE	11.761946	336.703067
RMSE	3.429569	18.349470
MAE	0.702970	8.051065
R2	0.999873	0.996370
MAPR	0.001378	0.014772

XGB**Evaluation Metrics:**

	XGB Train OneHot	XGB Test OneHot
MSE	30.026445	65.719158
RMSE	5.479639	8.106735
MAE	2.533013	3.429232
R2	0.999675	0.999291
MAPR	0.004893	0.006648

```
{"objective": "reg:squarederror", "base_score": None, "booster": None, "colsample_bylevel": None, "colsample_bynode": None, "colsample_bytree": 0.75, "eval_metric": None, "gamma": 0.35, "gpu_id": None, "grow_policy": None, "interaction_constraints": None, "learning_rate": 0.06, "max_bin": None, "max_cat_threshold": None, "max_cat_to_onehot": None, "max_delta_step": None, "max_depth": 50, "max_leaves": None, "min_child_weight": 2, "monotone_constraints": None, "n_jobs": None, "num_parallel_tree": None, "predictor": None, "random_state": None, "reg_alpha": 30, "reg_lambda": 0.01, "sampling_method": None, "scale_pos_weight": None, "subsample": 0.7, "tree_method": "gpu_hist", "validate_parameters": None, "verbosity": None}
```

SVR

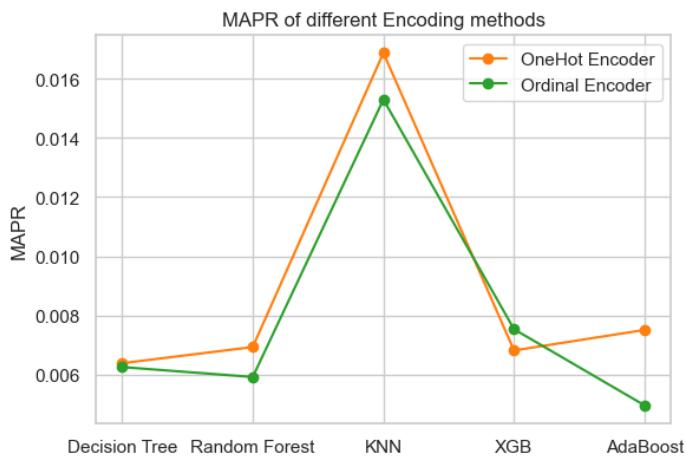
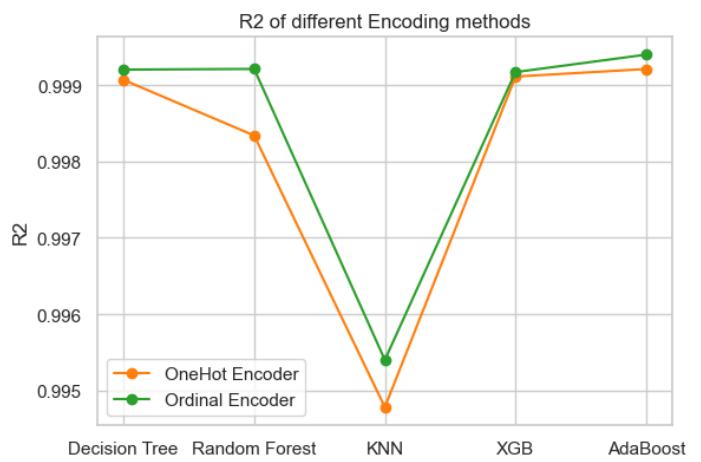
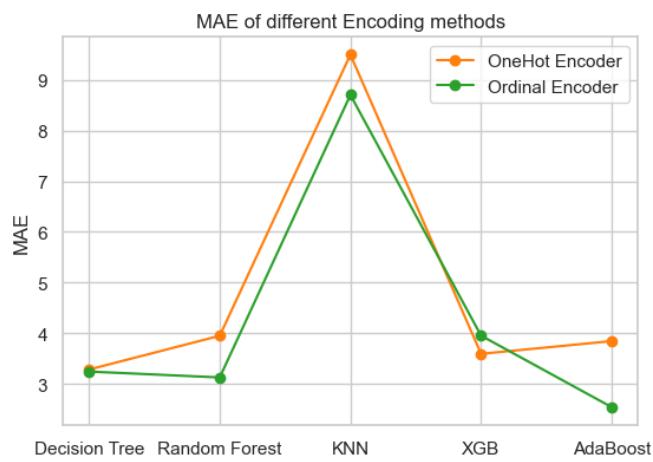
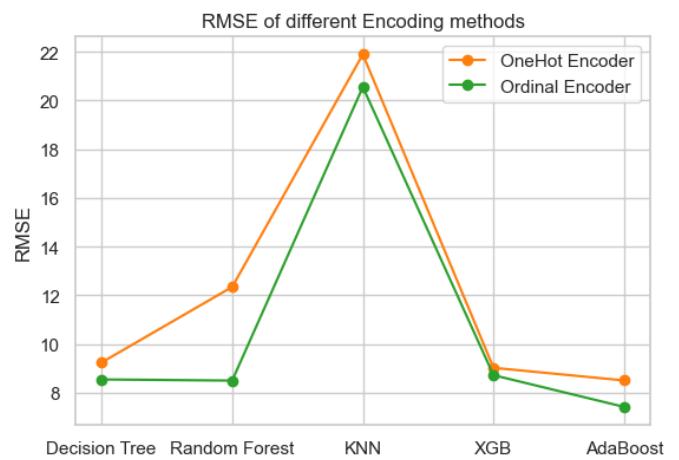
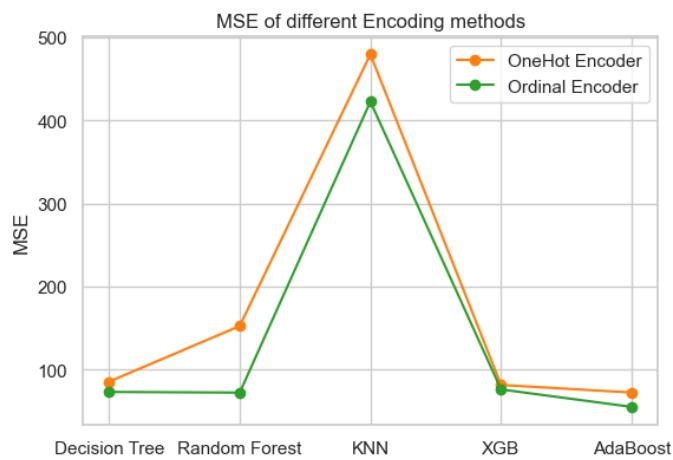
Evaluation Metrics:

	SVR Train OneHot	SVR Test OneHot
MSE	667.777238	970.165947
RMSE	25.841386	31.147487
MAE	9.413403	11.768888
R2	0.992775	0.989541
MAPR	0.015477	0.019455

ADA**Evaluation Metrics:**

	AdaBoost Train OneHot	AdaBoost Test OneHot
MSE	52.522797	78.499650
RMSE	7.247261	8.860003
MAE	3.231612	3.828675
R2	0.999432	0.999154
MAPR	0.006258	0.007509

	Decision Tree OneHot out of sample	Random Forest OneHot out of sample	KNN OneHot out of sample	XGB OneHot out of sample	SVR OneHot out of sample	AdaBoost OneHot out of sample
MSE	85.453642	152.434609	479.256187	81.498908	897.724242	72.393034
RMSE	9.244114	12.346441	21.891921	9.027675	29.962047	8.508410
MAE	3.280833	3.948012	9.504964	3.590986	12.000658	3.842671
R2	0.999069	0.998340	0.994781	0.999112	0.990223	0.999212
MAPR	0.006384	0.006931	0.016871	0.006816	0.019398	0.007510



With the additional consideration that less hyperparameter tuning was performed on models with OneHot encoding compared to Ordinal encoding, some of the performance differences could be attributed to this imbalance in optimization efforts.

OneHot encoding, while it can lead to a high dimensionality due to the creation of additional binary features, tends to perform better in algorithms such as XGB and AdaBoost as per the observed metrics. However, the hyperparameters for these models may not be as well-tuned as those using Ordinal encoding, potentially affecting the performance comparisons.

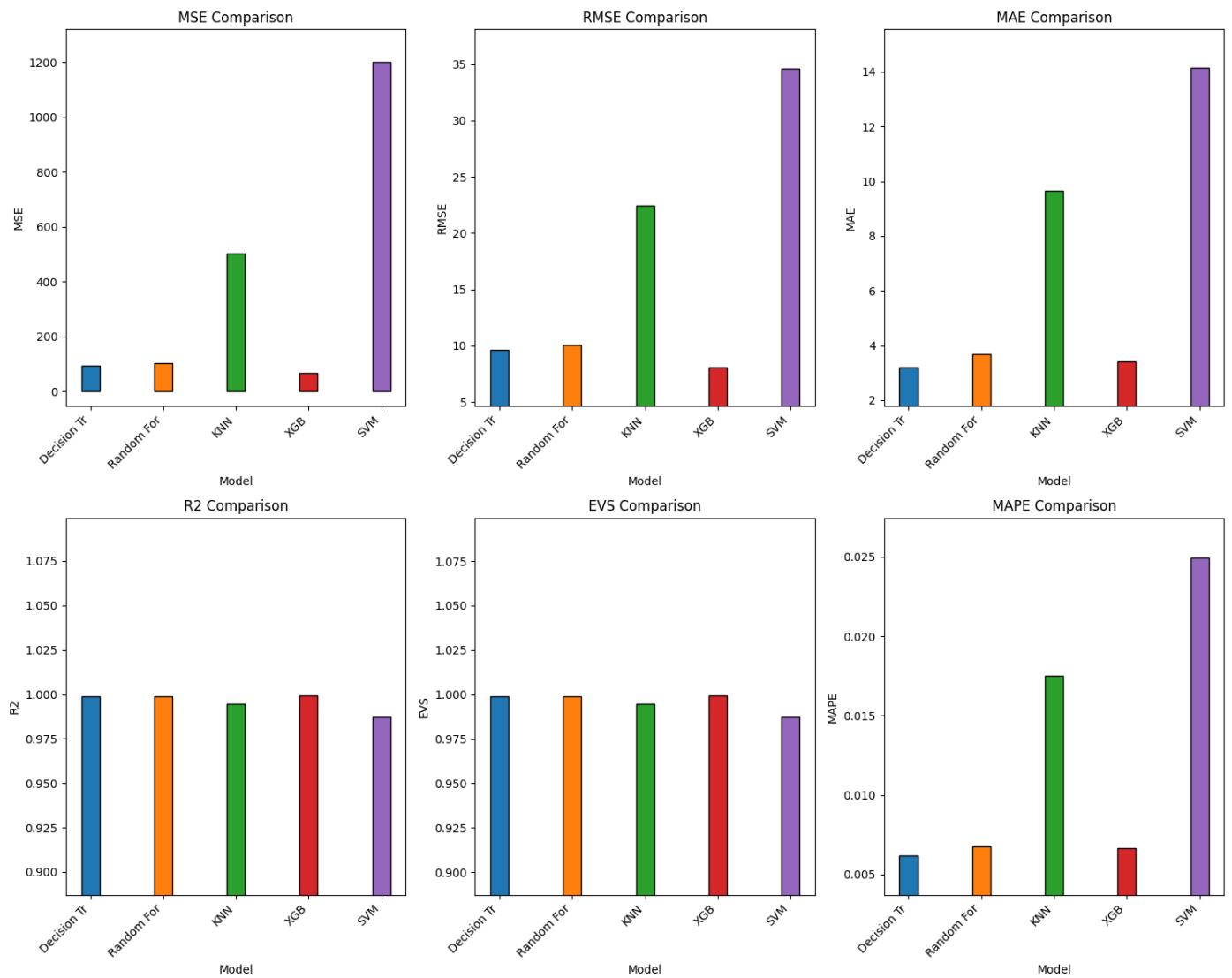
Ordinal encoding, on the other hand, reduces dimensionality and appears to perform better in Decision Trees and Random Forest models based on the metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These models may have been more fine-tuned, providing a more optimized performance.

The Support Vector Machine (SVM) model is notably sensitive to Ordinal encoding, experiencing a significant increase in error rates (MSE, RMSE, and MAE). This is likely because the inherent ranking in Ordinal encoding may distort the data space for SVM, impacting its ability to find an optimal hyperplane.

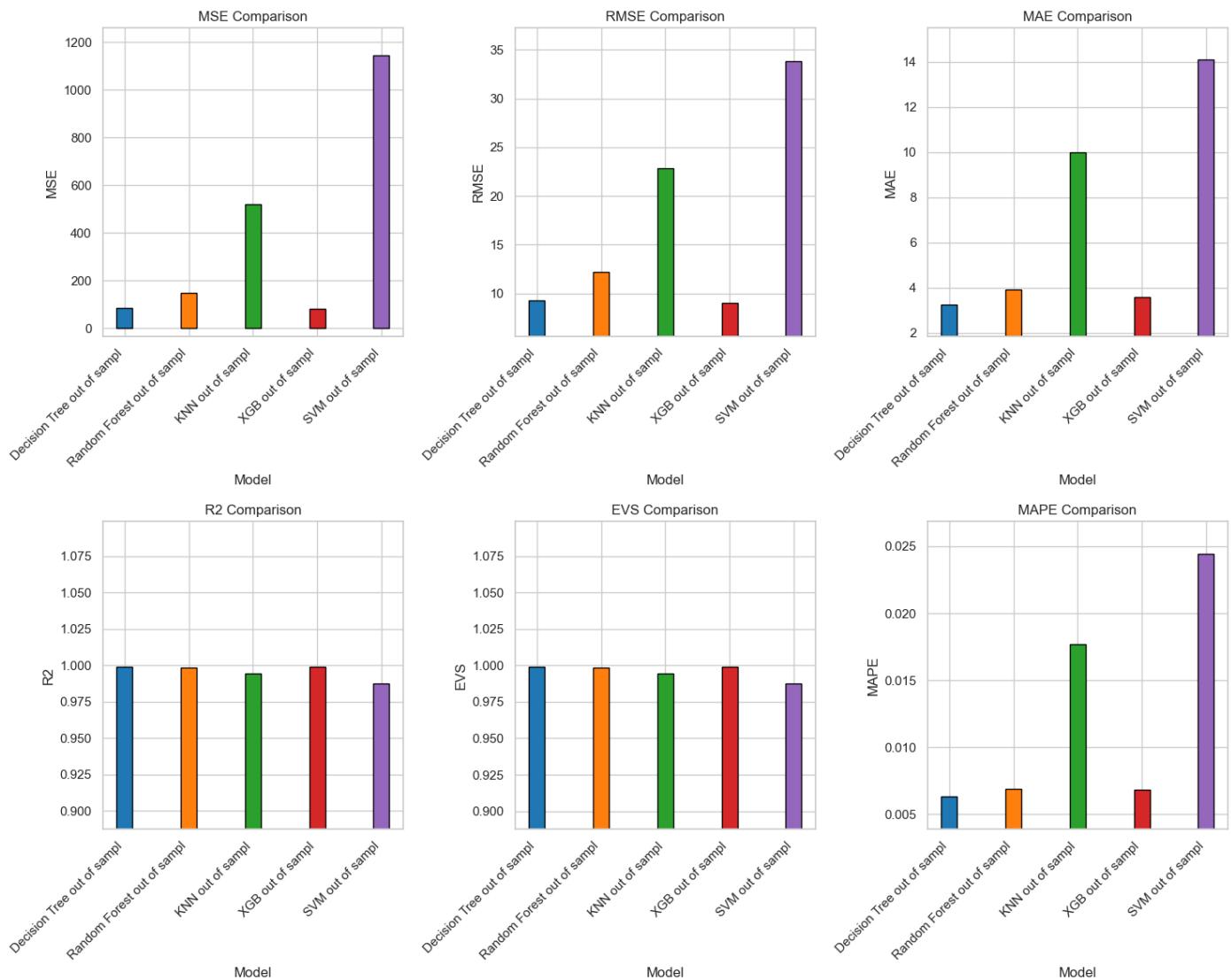
K-Nearest Neighbors (KNN) model is seemingly insensitive to the type of encoding used. However, the performance could potentially improve with more rigorous hyperparameter tuning in the OneHot encoded model.

Overall, it's essential to note that any performance comparison between the models should account for the potential influence of hyperparameter tuning. The observed differences may not solely be due to the choice of encoding method but also the level of optimization for each model. Thus, for a more accurate assessment, it would be beneficial to ensure equal hyperparameter tuning efforts for both OneHot- and Ordinal-encoded models.

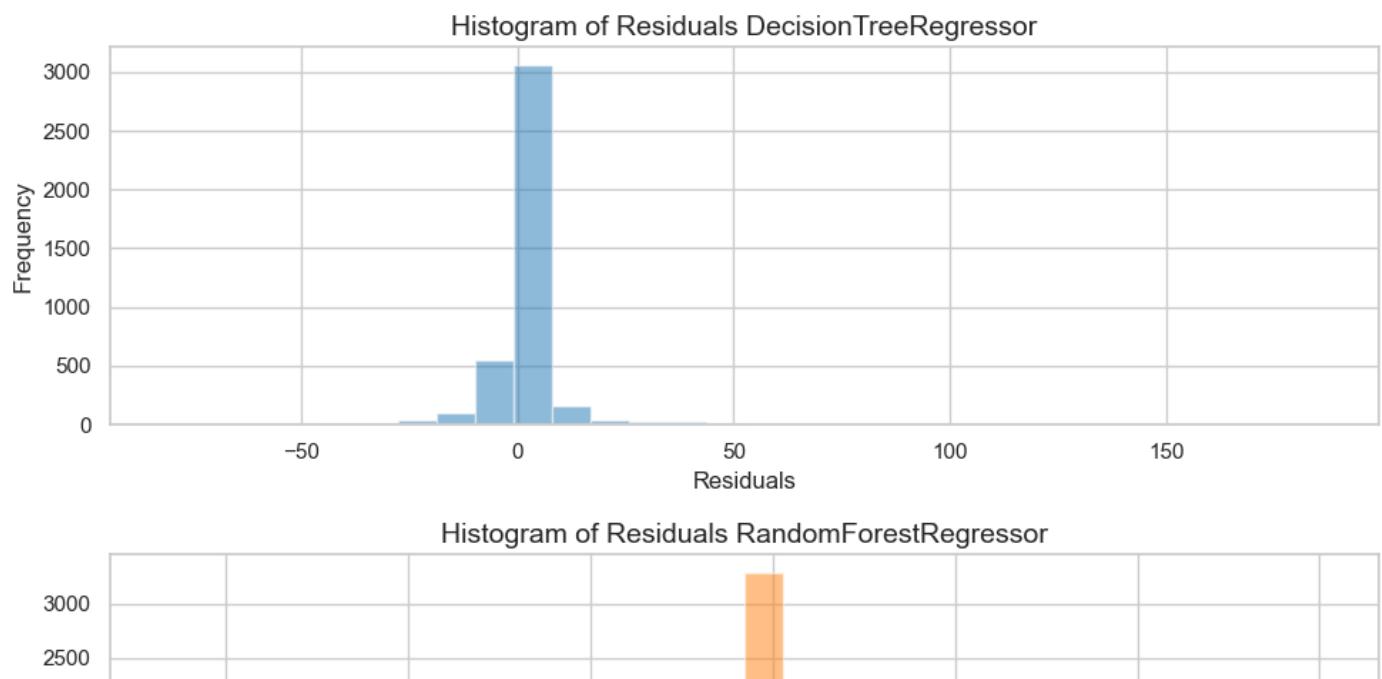
A2 Bar plots Test Performance

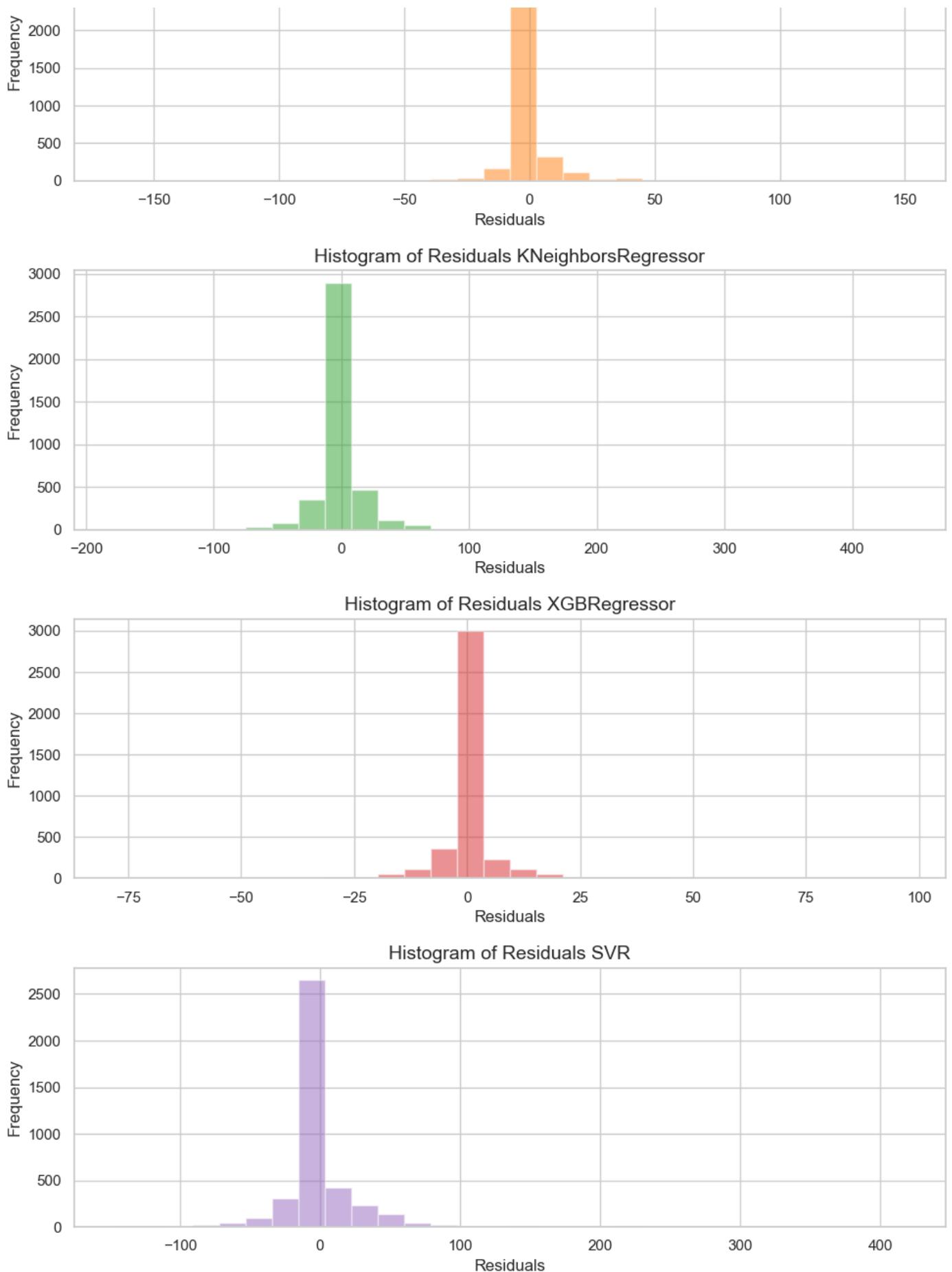


A3 Bar plots Out of sample Performance

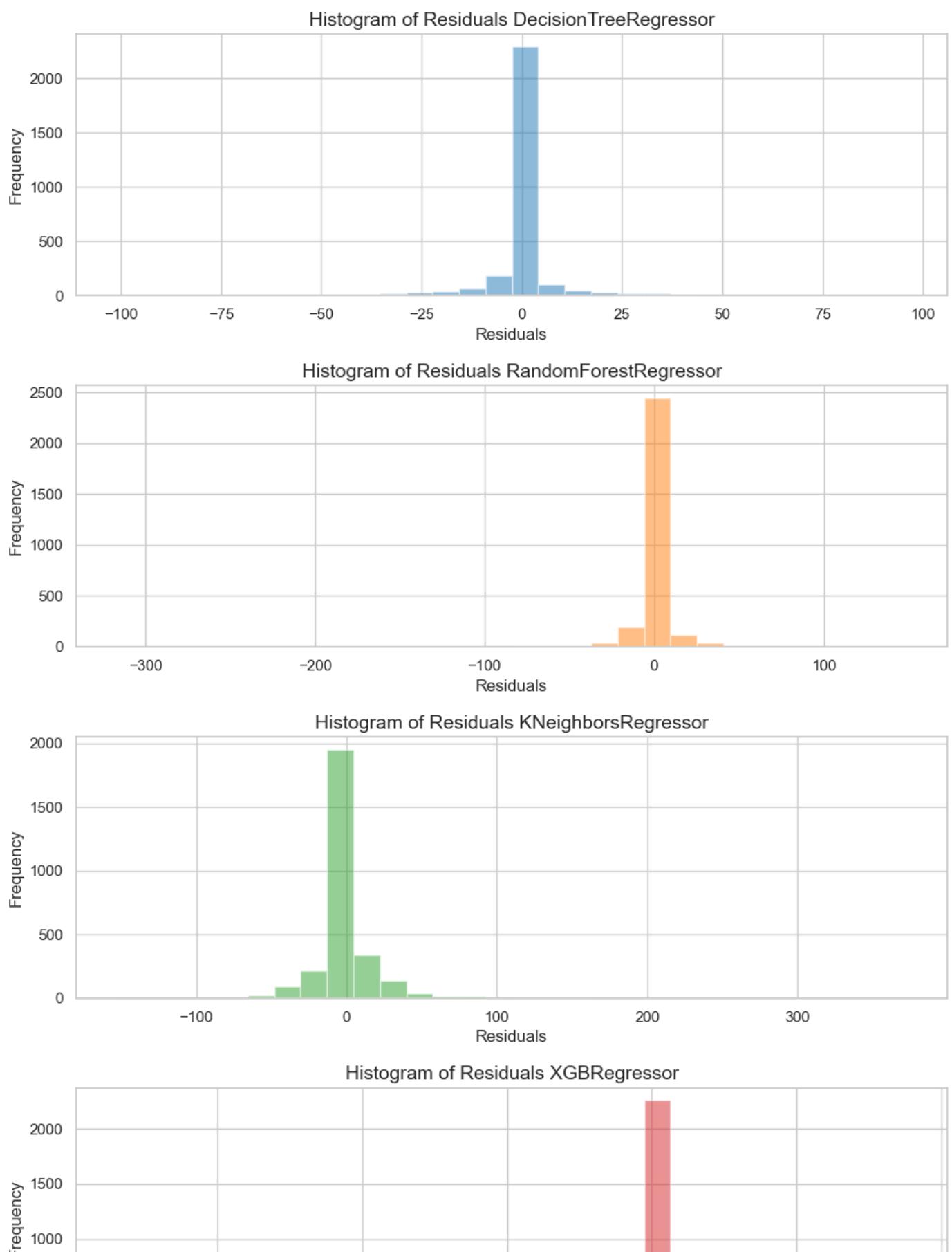


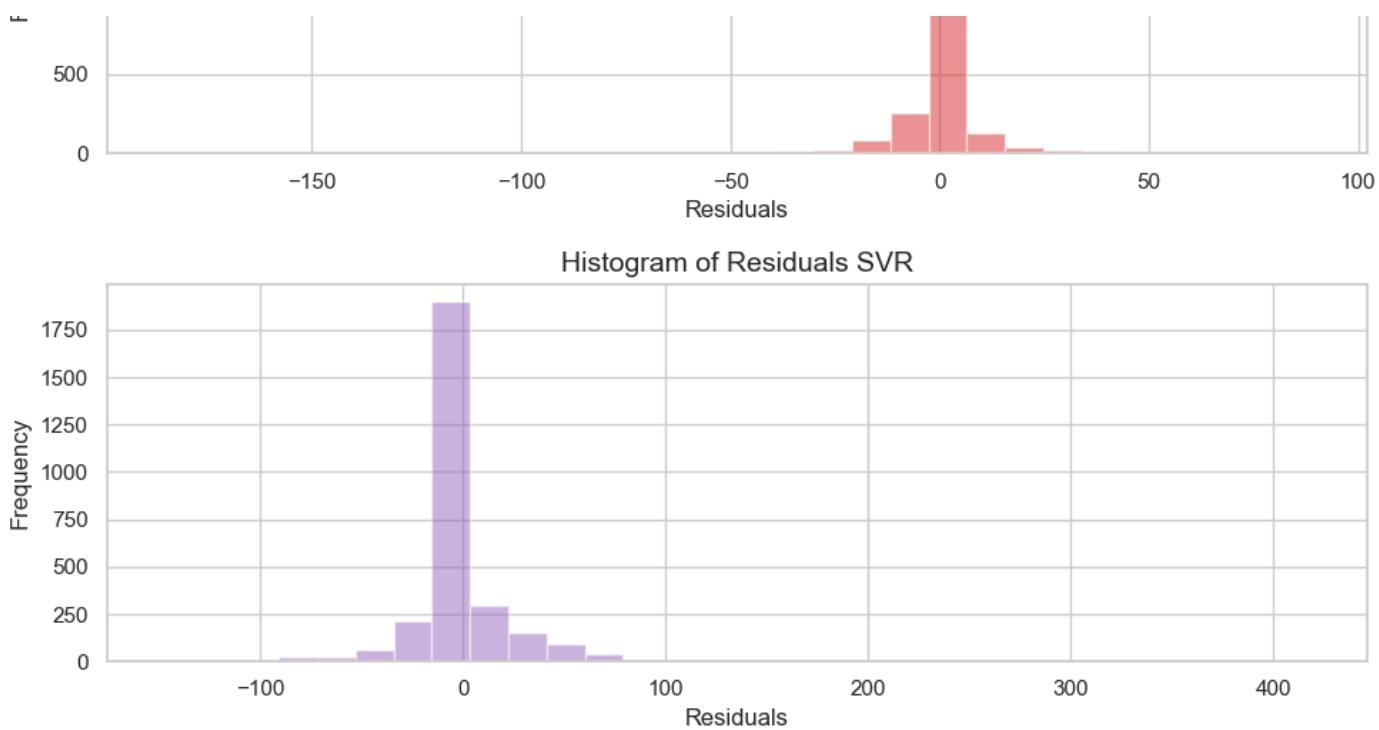
A4 Histogram of residuals test performance





A5 Histogram of residuals out of sample performance





A6 Light models (reduced complexity)

Before switching to ordinal encoding, we used OneHot encoding, which changed the results of the feature importance analysis. In addition, OneHot encoding increases the complexity of a model immensely, because it uses $n-1$ columns for n values inside a categorical feature.

For the OneHot encoded features, the results were, that the model of the car was not very significant. So we tried to build models with reduced complexity, the light models.

This section shows the building and evaluation of the light models and also shows the effect of leaving out the models in the model building process.

```
Index(['brand', 'gear', 'fuel'], dtype='object')
```

A6.1 Best two performing full models

A6.1.2 Random Forest

CPU usage: 0.6%
Memory usage: svmem(total=34276040704, available=21338042368, percent=37.7, used=12937998336, free=21338042368)%
Fitting 10 folds for each of 90 candidates, totalling 900 fits

Evaluation Metrics:

	Train Set	Test Set
MSE	209.742431	406.373134
RMSE	14.482487	20.158699
MAE	4.266952	6.065162
R2	0.997731	0.995619
EVS	0.997731	0.995619
MAPE	0.007943	0.011467

Parameters

regressor__ccp_alpha	0.018405
regressor__min_samples_leaf	2.000000
regressor__min_samples_split	10.000000

Computation time: 10.009563684463501

Evaluation Metrics:

	Decision Tree Train	Decision Tree Test
MSE	209.742431	413.383357
RMSE	14.482487	20.331831
MAE	4.266952	6.098616
R2	0.997731	0.995543
EVS	0.997731	0.995543
MAPE	0.007943	0.011526

A6.1.2 Random Forest

CPU usage: 2.3%
Memory usage: svmem(total=34276040704, available=21407842304, percent=37.5, used=12868198400, free=21407842304)%
Fitting 10 folds for each of 90 candidates, totalling 900 fits
CPU usage: 66.8%
Memory usage: svmem(total=34276040704, available=19473563648, percent=43.2, used=14802477056, free=19473563648)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=19033309184, percent=44.5, used=15242731520, free=19033309184)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=19707826176, percent=42.5, used=14568214528, free=19707826176)%
CPU usage: 78.7%
Memory usage: svmem(total=34276040704, available=19476279296, percent=43.2, used=14799761408, free=19476279296)%
CPU usage: 79.1%
Memory usage: svmem(total=34276040704, available=19850948608, percent=42.1, used=14425092096, free=19850948608)%
CPU usage: 86.1%
Memory usage: svmem(total=34276040704, available=19570376704, percent=42.9, used=14705664000, free=19570376704)%
CPU usage: 90.9%

Memory usage: svmem(total=34276040704, available=19687137280, percent=42.6, used=14588903424, free=19687137280)%
CPU usage: 90.1%
Memory usage: svmem(total=34276040704, available=19596967936, percent=42.8, used=14679072768, free=19596967936)%
CPU usage: 90.1%
Memory usage: svmem(total=34276040704, available=19624620032, percent=42.7, used=14651420672, free=19624620032)%
CPU usage: 90.7%
Memory usage: svmem(total=34276040704, available=19669188608, percent=42.6, used=14606852096, free=19669188608)%
CPU usage: 90.6%
Memory usage: svmem(total=34276040704, available=19822325760, percent=42.2, used=14453714944, free=19822325760)%
CPU usage: 90.0%
Memory usage: svmem(total=34276040704, available=19732127744, percent=42.4, used=14543912960, free=19732127744)%
CPU usage: 91.0%
Memory usage: svmem(total=34276040704, available=19342852096, percent=43.6, used=14933188608, free=19342852096)%
CPU usage: 91.0%
Memory usage: svmem(total=34276040704, available=19709231104, percent=42.5, used=14566809600, free=19709231104)%
CPU usage: 90.3%
Memory usage: svmem(total=34276040704, available=19865571328, percent=42.0, used=14410469376, free=19865571328)%
CPU usage: 92.3%
Memory usage: svmem(total=34276040704, available=20052123648, percent=41.5, used=14223917056, free=20052123648)%
CPU usage: 78.9%
Memory usage: svmem(total=34276040704, available=20296765440, percent=40.8, used=13979275264, free=20296765440)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=20249853952, percent=40.9, used=14026186752, free=20249853952)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=20059172864, percent=41.5, used=14216867840, free=20059172864)%
CPU usage: 78.3%
Memory usage: svmem(total=34276040704, available=20233490432, percent=41.0, used=14042550272, free=20233490432)%
CPU usage: 77.8%
Memory usage: svmem(total=34276040704, available=20220747776, percent=41.0, used=14055292928, free=20220747776)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=20195139584, percent=41.1, used=14080901120, free=20195139584)%
CPU usage: 78.1%
Memory usage: svmem(total=34276040704, available=20408881152, percent=40.5, used=13867159552, free=20408881152)%
CPU usage: 82.8%
Memory usage: svmem(total=34276040704, available=20231221248, percent=41.0, used=14044819456, free=20231221248)%
CPU usage: 91.0%

Memory usage: svmem(total=34276040704, available=20143226880, percent=41.2, used=14132813824, free=20143226880)%
CPU usage: 90.2%
Memory usage: svmem(total=34276040704, available=20067160064, percent=41.5, used=14208880640, free=20067160064)%
CPU usage: 90.2%
Memory usage: svmem(total=34276040704, available=19744190464, percent=42.4, used=14531850240, free=19744190464)%
CPU usage: 90.8%
Memory usage: svmem(total=34276040704, available=20068962304, percent=41.4, used=14207078400, free=20068962304)%
CPU usage: 90.2%
Memory usage: svmem(total=34276040704, available=19849539584, percent=42.1, used=14426501120, free=19849539584)%
CPU usage: 89.9%
Memory usage: svmem(total=34276040704, available=20253585408, percent=40.9, used=14022455296, free=20253585408)%
CPU usage: 89.8%
Memory usage: svmem(total=34276040704, available=20180373504, percent=41.1, used=14095667200, free=20180373504)%
CPU usage: 90.4%
Memory usage: svmem(total=34276040704, available=20047409152, percent=41.5, used=14228631552, free=20047409152)%
CPU usage: 90.0%
Memory usage: svmem(total=34276040704, available=19806359552, percent=42.2, used=14469681152, free=19806359552)%
CPU usage: 95.2%
Memory usage: svmem(total=34276040704, available=20124291072, percent=41.3, used=14151749632, free=20124291072)%
CPU usage: 79.8%
Memory usage: svmem(total=34276040704, available=20271095808, percent=40.9, used=14004944896, free=20271095808)%
CPU usage: 77.0%
Memory usage: svmem(total=34276040704, available=20245766144, percent=40.9, used=14030274560, free=20245766144)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=20235542528, percent=41.0, used=14040498176, free=20235542528)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=20027699200, percent=41.6, used=14248341504, free=20027699200)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=19990892544, percent=41.7, used=14285148160, free=19990892544)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=20195328000, percent=41.1, used=14080712704, free=20195328000)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=20077551616, percent=41.4, used=14198489088, free=20077551616)%
CPU usage: 81.4%
Memory usage: svmem(total=34276040704, available=20067045376, percent=41.5, used=14208995328, free=20067045376)%
CPU usage: 91.8%

Memory usage: svmem(total=34276040704, available=20185923584, percent=41.1, used=14090117120, free=20185923584)%
CPU usage: 91.2%
Memory usage: svmem(total=34276040704, available=20183048192, percent=41.1, used=14092992512, free=20183048192)%
CPU usage: 90.4%
Memory usage: svmem(total=34276040704, available=20040790016, percent=41.5, used=14235250688, free=20040790016)%
CPU usage: 90.7%
Memory usage: svmem(total=34276040704, available=20254322688, percent=40.9, used=14021718016, free=20254322688)%
CPU usage: 89.3%
Memory usage: svmem(total=34276040704, available=20035432448, percent=41.5, used=14240608256, free=20035432448)%
CPU usage: 90.4%
Memory usage: svmem(total=34276040704, available=20261806080, percent=40.9, used=14014234624, free=20261806080)%
CPU usage: 90.1%
Memory usage: svmem(total=34276040704, available=20030644224, percent=41.6, used=14245396480, free=20030644224)%
CPU usage: 89.9%
Memory usage: svmem(total=34276040704, available=20170878976, percent=41.2, used=14105161728, free=20170878976)%
CPU usage: 89.9%
Memory usage: svmem(total=34276040704, available=20220239872, percent=41.0, used=14055800832, free=20220239872)%
CPU usage: 90.5%
Memory usage: svmem(total=34276040704, available=20118073344, percent=41.3, used=14157967360, free=20118073344)%
CPU usage: 81.9%
Memory usage: svmem(total=34276040704, available=20067299328, percent=41.5, used=14208741376, free=20067299328)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=20013006848, percent=41.6, used=14263033856, free=20013006848)%
CPU usage: 78.2%
Memory usage: svmem(total=34276040704, available=19962478592, percent=41.8, used=14313562112, free=19962478592)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=20127072256, percent=41.3, used=14148968448, free=20127072256)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=20152082432, percent=41.2, used=14123958272, free=20152082432)%
CPU usage: 79.2%
Memory usage: svmem(total=34276040704, available=20228362240, percent=41.0, used=14047678464, free=20228362240)%
CPU usage: 78.7%
Memory usage: svmem(total=34276040704, available=20228370432, percent=41.0, used=14047670272, free=20228370432)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=19943337984, percent=41.8, used=14332702720, free=19943337984)%
CPU usage: 89.9%

Memory usage: svmem(total=34276040704, available=20185042944, percent=41.1, used=14090997760, free=20185042944)%
CPU usage: 91.5%
Memory usage: svmem(total=34276040704, available=20447232000, percent=40.3, used=13828808704, free=20447232000)%
CPU usage: 89.6%
Memory usage: svmem(total=34276040704, available=20283809792, percent=40.8, used=13992230912, free=20283809792)%
CPU usage: 90.2%
Memory usage: svmem(total=34276040704, available=20376051712, percent=40.6, used=13899988992, free=20376051712)%
CPU usage: 90.5%
Memory usage: svmem(total=34276040704, available=20075679744, percent=41.4, used=14200360960, free=20075679744)%
CPU usage: 90.8%
Memory usage: svmem(total=34276040704, available=19962224640, percent=41.8, used=14313816064, free=19962224640)%
CPU usage: 89.8%
Memory usage: svmem(total=34276040704, available=19844702208, percent=42.1, used=14431338496, free=19844702208)%
CPU usage: 90.5%
Memory usage: svmem(total=34276040704, available=20368482304, percent=40.6, used=13907558400, free=20368482304)%
CPU usage: 90.6%
Memory usage: svmem(total=34276040704, available=20416143360, percent=40.4, used=13859897344, free=20416143360)%
CPU usage: 90.4%
Memory usage: svmem(total=34276040704, available=20333285376, percent=40.7, used=13942755328, free=20333285376)%
CPU usage: 51.0%
Memory usage: svmem(total=34276040704, available=20523012096, percent=40.1, used=13753028608, free=20523012096)%

Evaluation Metrics:

	Train Set	Test Set
MSE	393.864941	460.062865
RMSE	19.846031	21.449076
MAE	7.408231	8.393508
R2	0.995739	0.995040
EVS	0.995739	0.995041
MAPE	0.012797	0.014846

Parameters

regressor__max_depth	80
regressor__min_samples_leaf	6
regressor__min_samples_split	12
regressor__n_estimators	500

Computation time: 720.7719922065735

Evaluation Metrics:

	Random Forest Train	Random Forest Test
MSE	385.868909	451.555470
RMSE	19.643546	21.249835
MAE	7.373744	8.347957
R2	0.995825	0.995132
EVS	0.995825	0.995133
MAPE	0.012751	0.014797

A6.1.3 XGB

CPU usage: 2.5%
Memory usage: svmem(total=34276040704, available=20512382976, percent=40.2, used=13763657728, free=20512382976)%
Fitting 10 folds for each of 90 candidates, totalling 900 fits
CPU usage: 82.1%
Memory usage: svmem(total=34276040704, available=17220538368, percent=49.8, used=17055502336, free=17220538368)%
CPU usage: 86.4%
Memory usage: svmem(total=34276040704, available=17124716544, percent=50.0, used=17151324160, free=17124716544)%
CPU usage: 85.8%
Memory usage: svmem(total=34276040704, available=17124773888, percent=50.0, used=17151266816, free=17124773888)%
CPU usage: 84.9%
Memory usage: svmem(total=34276040704, available=17113300992, percent=50.1, used=17162739712, free=17113300992)%
CPU usage: 89.7%
Memory usage: svmem(total=34276040704, available=17049661440, percent=50.3, used=17226379264, free=17049661440)%
CPU usage: 86.7%
Memory usage: svmem(total=34276040704, available=17098117120, percent=50.1, used=17177923584, free=17098117120)%
CPU usage: 85.6%
Memory usage: svmem(total=34276040704, available=17045811200, percent=50.3, used=17230229504, free=17045811200)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=17033105408, percent=50.3, used=17242935296, free=17033105408)%
CPU usage: 70.9%
Memory usage: svmem(total=34276040704, available=17028702208, percent=50.3, used=17247338496, free=17028702208)%
CPU usage: 74.6%
Memory usage: svmem(total=34276040704, available=17039785984, percent=50.3, used=17236254720, free=17039785984)%
CPU usage: 72.2%
Memory usage: svmem(total=34276040704, available=17023582208, percent=50.3, used=17252458496, free=17023582208)%
CPU usage: 70.4%
Memory usage: svmem(total=34276040704, available=17029574656, percent=50.3, used=17246466048, free=17029574656)%
CPU usage: 70.9%
Memory usage: svmem(total=34276040704, available=16978833408, percent=50.5, used=1

7297207296, free=16978833408)%
CPU usage: 70.6%
Memory usage: svmem(total=34276040704, available=16977989632, percent=50.5, used=17298051072, free=16977989632)%
CPU usage: 81.0%
Memory usage: svmem(total=34276040704, available=16956149760, percent=50.5, used=17319890944, free=16956149760)%
CPU usage: 77.0%
Memory usage: svmem(total=34276040704, available=17007669248, percent=50.4, used=17268371456, free=17007669248)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=16908419072, percent=50.7, used=17367621632, free=16908419072)%
CPU usage: 88.1%
Memory usage: svmem(total=34276040704, available=16920866816, percent=50.6, used=17355173888, free=16920866816)%
CPU usage: 87.4%
Memory usage: svmem(total=34276040704, available=16868937728, percent=50.8, used=17407102976, free=16868937728)%
CPU usage: 85.4%
Memory usage: svmem(total=34276040704, available=17024434176, percent=50.3, used=17251606528, free=17024434176)%
CPU usage: 88.1%
Memory usage: svmem(total=34276040704, available=17058463744, percent=50.2, used=17217576960, free=17058463744)%
CPU usage: 84.1%
Memory usage: svmem(total=34276040704, available=17037733888, percent=50.3, used=17238306816, free=17037733888)%
CPU usage: 85.4%
Memory usage: svmem(total=34276040704, available=17065803776, percent=50.2, used=17210236928, free=17065803776)%
CPU usage: 86.1%
Memory usage: svmem(total=34276040704, available=17046376448, percent=50.3, used=17229664256, free=17046376448)%
CPU usage: 86.1%
Memory usage: svmem(total=34276040704, available=17023664128, percent=50.3, used=17252376576, free=17023664128)%
CPU usage: 87.2%
Memory usage: svmem(total=34276040704, available=17024282624, percent=50.3, used=17251758080, free=17024282624)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=16973553664, percent=50.5, used=17302487040, free=16973553664)%
CPU usage: 70.5%
Memory usage: svmem(total=34276040704, available=16962236416, percent=50.5, used=17313804288, free=16962236416)%
CPU usage: 82.4%
Memory usage: svmem(total=34276040704, available=18129977344, percent=47.1, used=16146063360, free=18129977344)%
CPU usage: 83.8%
Memory usage: svmem(total=34276040704, available=18458038272, percent=46.1, used=15818002432, free=18458038272)%
CPU usage: 71.3%
Memory usage: svmem(total=34276040704, available=18475528192, percent=46.1, used=1

5800512512, free=18475528192)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=18465673216, percent=46.1, used=15810367488, free=18465673216)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=18427834368, percent=46.2, used=15848206336, free=18427834368)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=18418520064, percent=46.3, used=15857520640, free=18418520064)%
CPU usage: 90.2%
Memory usage: svmem(total=34276040704, available=18331205632, percent=46.5, used=15944835072, free=18331205632)%
CPU usage: 95.5%
Memory usage: svmem(total=34276040704, available=18263109632, percent=46.7, used=16012931072, free=18263109632)%
CPU usage: 92.8%
Memory usage: svmem(total=34276040704, available=18239463424, percent=46.8, used=16036577280, free=18239463424)%
CPU usage: 85.5%
Memory usage: svmem(total=34276040704, available=18224914432, percent=46.8, used=16051126272, free=18224914432)%
CPU usage: 96.9%
Memory usage: svmem(total=34276040704, available=18250129408, percent=46.8, used=16025911296, free=18250129408)%
CPU usage: 97.3%
Memory usage: svmem(total=34276040704, available=18238070784, percent=46.8, used=16037969920, free=18238070784)%
CPU usage: 98.8%
Memory usage: svmem(total=34276040704, available=18241871872, percent=46.8, used=16034168832, free=18241871872)%
CPU usage: 99.9%
Memory usage: svmem(total=34276040704, available=18234613760, percent=46.8, used=16041426944, free=18234613760)%
CPU usage: 99.8%
Memory usage: svmem(total=34276040704, available=18237456384, percent=46.8, used=16038584320, free=18237456384)%
CPU usage: 99.8%
Memory usage: svmem(total=34276040704, available=18348056576, percent=46.5, used=15927984128, free=18348056576)%
CPU usage: 100.0%
Memory usage: svmem(total=34276040704, available=17965092864, percent=47.6, used=16310947840, free=17965092864)%
CPU usage: 92.6%
Memory usage: svmem(total=34276040704, available=17920856064, percent=47.7, used=16355184640, free=17920856064)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=17962659840, percent=47.6, used=16313380864, free=17962659840)%
CPU usage: 72.9%
Memory usage: svmem(total=34276040704, available=18003009536, percent=47.5, used=16273031168, free=18003009536)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17957928960, percent=47.6, used=1

6318111744, free=17957928960)%
CPU usage: 74.6%
Memory usage: svmem(total=34276040704, available=17902215168, percent=47.8, used=1
6373825536, free=17902215168)%
CPU usage: 83.7%
Memory usage: svmem(total=34276040704, available=17877594112, percent=47.8, used=1
6398446592, free=17877594112)%
CPU usage: 80.0%
Memory usage: svmem(total=34276040704, available=17866792960, percent=47.9, used=1
6409247744, free=17866792960)%
CPU usage: 81.6%
Memory usage: svmem(total=34276040704, available=17888780288, percent=47.8, used=1
6387260416, free=17888780288)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=17920634880, percent=47.7, used=1
6355405824, free=17920634880)%
CPU usage: 76.7%
Memory usage: svmem(total=34276040704, available=17929080832, percent=47.7, used=1
6346959872, free=17929080832)%
CPU usage: 81.7%
Memory usage: svmem(total=34276040704, available=17908060160, percent=47.8, used=1
6367980544, free=17908060160)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17910124544, percent=47.7, used=1
6365916160, free=17910124544)%
CPU usage: 74.3%
Memory usage: svmem(total=34276040704, available=17877467136, percent=47.8, used=1
6398573568, free=17877467136)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=17935433728, percent=47.7, used=1
6340606976, free=17935433728)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=17950752768, percent=47.6, used=1
6325287936, free=17950752768)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=17908170752, percent=47.8, used=1
6367869952, free=17908170752)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=17892216832, percent=47.8, used=1
6383823872, free=17892216832)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=17904840704, percent=47.8, used=1
6371200000, free=17904840704)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17882853376, percent=47.8, used=1
6393187328, free=17882853376)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17855602688, percent=47.9, used=1
6420438016, free=17855602688)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17878167552, percent=47.8, used=1
6397873152, free=17878167552)%
CPU usage: 84.3%
Memory usage: svmem(total=34276040704, available=17869758464, percent=47.9, used=1

6406282240, free=17869758464)%
CPU usage: 84.4%
Memory usage: svmem(total=34276040704, available=17941762048, percent=47.7, used=16334278656, free=17941762048)%
CPU usage: 79.1%
Memory usage: svmem(total=34276040704, available=17951076352, percent=47.6, used=16324964352, free=17951076352)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=17904160768, percent=47.8, used=16371879936, free=17904160768)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=17931137024, percent=47.7, used=16344903680, free=17931137024)%
A worker stopped while some jobs were given to the executor. This can be caused by a too short worker timeout or by a memory leak.
CPU usage: 81.9%
Memory usage: svmem(total=34276040704, available=18218610688, percent=46.8, used=16057430016, free=18218610688)%
CPU usage: 77.8%
Memory usage: svmem(total=34276040704, available=17888964608, percent=47.8, used=16387076096, free=17888964608)%
CPU usage: 77.4%
Memory usage: svmem(total=34276040704, available=17950658560, percent=47.6, used=16325382144, free=17950658560)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=17970810880, percent=47.6, used=16305229824, free=17970810880)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=17952088064, percent=47.6, used=16323952640, free=17952088064)%
CPU usage: 78.3%
Memory usage: svmem(total=34276040704, available=17948573696, percent=47.6, used=16327467008, free=17948573696)%
CPU usage: 86.0%
Memory usage: svmem(total=34276040704, available=18045554688, percent=47.4, used=16230486016, free=18045554688)%
CPU usage: 82.1%
Memory usage: svmem(total=34276040704, available=17924358144, percent=47.7, used=16351682560, free=17924358144)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=17989771264, percent=47.5, used=16286269440, free=17989771264)%
CPU usage: 84.3%
Memory usage: svmem(total=34276040704, available=17982279680, percent=47.5, used=16293761024, free=17982279680)%
CPU usage: 79.4%
Memory usage: svmem(total=34276040704, available=17941229568, percent=47.7, used=16334811136, free=17941229568)%
CPU usage: 79.6%
Memory usage: svmem(total=34276040704, available=18014404608, percent=47.4, used=16261636096, free=18014404608)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=18002509824, percent=47.5, used=1

6273530880, free=18002509824)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=17962876928, percent=47.6, used=16313163776, free=17962876928)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=17946898432, percent=47.6, used=16329142272, free=17946898432)%
CPU usage: 79.8%
Memory usage: svmem(total=34276040704, available=18240114688, percent=46.8, used=16035926016, free=18240114688)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=18203783168, percent=46.9, used=16072257536, free=18203783168)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=18185482240, percent=46.9, used=16090558464, free=18185482240)%
CPU usage: 85.7%
Memory usage: svmem(total=34276040704, available=18207514624, percent=46.9, used=16068526080, free=18207514624)%
CPU usage: 81.7%
Memory usage: svmem(total=34276040704, available=18217865216, percent=46.8, used=16058175488, free=18217865216)%
CPU usage: 79.3%
Memory usage: svmem(total=34276040704, available=18298212352, percent=46.6, used=15977828352, free=18298212352)%
CPU usage: 84.1%
Memory usage: svmem(total=34276040704, available=18220961792, percent=46.8, used=16055078912, free=18220961792)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18255261696, percent=46.7, used=16020779008, free=18255261696)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=18224631808, percent=46.8, used=16051408896, free=18224631808)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=18237448192, percent=46.8, used=16038592512, free=18237448192)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=18204033024, percent=46.9, used=16072007680, free=18204033024)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=18233786368, percent=46.8, used=16042254336, free=18233786368)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=18208927744, percent=46.9, used=16067112960, free=18208927744)%
CPU usage: 83.8%
Memory usage: svmem(total=34276040704, available=18276757504, percent=46.7, used=15999283200, free=18276757504)%
CPU usage: 79.2%
Memory usage: svmem(total=34276040704, available=18284756992, percent=46.7, used=15991283712, free=18284756992)%
CPU usage: 80.7%
Memory usage: svmem(total=34276040704, available=18264657920, percent=46.7, used=1

6011382784, free=18264657920)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=18270097408, percent=46.7, used=1
6005943296, free=18270097408)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=18283286528, percent=46.7, used=1
5992754176, free=18283286528)%
CPU usage: 74.5%
Memory usage: svmem(total=34276040704, available=18227167232, percent=46.8, used=1
6048873472, free=18227167232)%
CPU usage: 73.5%
Memory usage: svmem(total=34276040704, available=18187505664, percent=46.9, used=1
6088535040, free=18187505664)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=18143662080, percent=47.1, used=1
6132378624, free=18143662080)%
CPU usage: 74.1%
Memory usage: svmem(total=34276040704, available=18185367552, percent=46.9, used=1
6090673152, free=18185367552)%
CPU usage: 73.0%
Memory usage: svmem(total=34276040704, available=18200518656, percent=46.9, used=1
6075522048, free=18200518656)%
CPU usage: 72.5%
Memory usage: svmem(total=34276040704, available=18172383232, percent=47.0, used=1
6103657472, free=18172383232)%
CPU usage: 79.4%
Memory usage: svmem(total=34276040704, available=18166521856, percent=47.0, used=1
6109518848, free=18166521856)%
CPU usage: 78.1%
Memory usage: svmem(total=34276040704, available=18161020928, percent=47.0, used=1
6115019776, free=18161020928)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=18133905408, percent=47.1, used=1
6142135296, free=18133905408)%
CPU usage: 79.1%
Memory usage: svmem(total=34276040704, available=18062168064, percent=47.3, used=1
6213872640, free=18062168064)%
CPU usage: 78.2%
Memory usage: svmem(total=34276040704, available=18075996160, percent=47.3, used=1
6200044544, free=18075996160)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=18078593024, percent=47.3, used=1
6197447680, free=18078593024)%
CPU usage: 79.4%
Memory usage: svmem(total=34276040704, available=18065768448, percent=47.3, used=1
6210272256, free=18065768448)%
CPU usage: 83.3%
Memory usage: svmem(total=34276040704, available=18190163968, percent=46.9, used=1
6085876736, free=18190163968)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=18182934528, percent=47.0, used=1
6093106176, free=18182934528)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=18166951936, percent=47.0, used=1

6109088768, free=18166951936)%
CPU usage: 89.3%
Memory usage: svmem(total=34276040704, available=17576914944, percent=48.7, used=16699125760, free=17576914944)%
CPU usage: 89.4%
Memory usage: svmem(total=34276040704, available=17350836224, percent=49.4, used=16925204480, free=17350836224)%
CPU usage: 87.2%
Memory usage: svmem(total=34276040704, available=17374330880, percent=49.3, used=16901709824, free=17374330880)%
CPU usage: 85.0%
Memory usage: svmem(total=34276040704, available=17816907776, percent=48.0, used=16459132928, free=17816907776)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17853415424, percent=47.9, used=16422625280, free=17853415424)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=17848147968, percent=47.9, used=16427892736, free=17848147968)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17860083712, percent=47.9, used=16415956992, free=17860083712)%
CPU usage: 76.5%
Memory usage: svmem(total=34276040704, available=17889865728, percent=47.8, used=16386174976, free=17889865728)%
CPU usage: 74.2%
Memory usage: svmem(total=34276040704, available=17900810240, percent=47.8, used=16375230464, free=17900810240)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=17885085696, percent=47.8, used=16390955008, free=17885085696)%
CPU usage: 79.0%
Memory usage: svmem(total=34276040704, available=17921740800, percent=47.7, used=16354299904, free=17921740800)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=17910591488, percent=47.7, used=16365449216, free=17910591488)%
CPU usage: 84.3%
Memory usage: svmem(total=34276040704, available=17944985600, percent=47.6, used=16331055104, free=17944985600)%
CPU usage: 83.5%
Memory usage: svmem(total=34276040704, available=18290790400, percent=46.6, used=15985250304, free=18290790400)%
CPU usage: 83.8%
Memory usage: svmem(total=34276040704, available=18024824832, percent=47.4, used=16251215872, free=18024824832)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=18025766912, percent=47.4, used=16250273792, free=18025766912)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18010042368, percent=47.5, used=16265998336, free=18010042368)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=17984417792, percent=47.5, used=1

6291622912, free=17984417792)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17972506624, percent=47.6, used=16303534080, free=17972506624)%
CPU usage: 80.1%
Memory usage: svmem(total=34276040704, available=18002706432, percent=47.5, used=16273334272, free=18002706432)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=17971187712, percent=47.6, used=16304852992, free=17971187712)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17959481344, percent=47.6, used=16316559360, free=17959481344)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=17935089664, percent=47.7, used=16340951040, free=17935089664)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=18013814784, percent=47.4, used=16262225920, free=18013814784)%
CPU usage: 77.3%
Memory usage: svmem(total=34276040704, available=18049441792, percent=47.3, used=16226598912, free=18049441792)%
CPU usage: 80.1%
Memory usage: svmem(total=34276040704, available=17987473408, percent=47.5, used=16288567296, free=17987473408)%
CPU usage: 76.5%
Memory usage: svmem(total=34276040704, available=17966276608, percent=47.6, used=16309764096, free=17966276608)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=17991692288, percent=47.5, used=16284348416, free=17991692288)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=18003922944, percent=47.5, used=16272117760, free=18003922944)%
CPU usage: 85.2%
Memory usage: svmem(total=34276040704, available=17903595520, percent=47.8, used=16372445184, free=17903595520)%
CPU usage: 85.4%
Memory usage: svmem(total=34276040704, available=18016071680, percent=47.4, used=16259969024, free=18016071680)%
CPU usage: 80.8%
Memory usage: svmem(total=34276040704, available=18300530688, percent=46.6, used=15975510016, free=18300530688)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=18126499840, percent=47.1, used=16149540864, free=18126499840)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=18192990208, percent=46.9, used=16083050496, free=18192990208)%
CPU usage: 81.2%
Memory usage: svmem(total=34276040704, available=18215133184, percent=46.9, used=16060907520, free=18215133184)%
CPU usage: 81.3%
Memory usage: svmem(total=34276040704, available=18174676992, percent=47.0, used=1

6101363712, free=18174676992)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=18169032704, percent=47.0, used=1
6107008000, free=18169032704)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=18165358592, percent=47.0, used=1
6110682112, free=18165358592)%
CPU usage: 77.0%
Memory usage: svmem(total=34276040704, available=18152636416, percent=47.0, used=1
6123404288, free=18152636416)%
CPU usage: 83.8%
Memory usage: svmem(total=34276040704, available=18133299200, percent=47.1, used=1
6142741504, free=18133299200)%
CPU usage: 81.9%
Memory usage: svmem(total=34276040704, available=18114584576, percent=47.2, used=1
6161456128, free=18114584576)%
CPU usage: 80.3%
Memory usage: svmem(total=34276040704, available=18142232576, percent=47.1, used=1
6133808128, free=18142232576)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=18111553536, percent=47.2, used=1
6164487168, free=18111553536)%
CPU usage: 83.4%
Memory usage: svmem(total=34276040704, available=18406461440, percent=46.3, used=1
5869579264, free=18406461440)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=18174955520, percent=47.0, used=1
6101085184, free=18174955520)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18156863488, percent=47.0, used=1
6119177216, free=18156863488)%
CPU usage: 82.1%
Memory usage: svmem(total=34276040704, available=18087424000, percent=47.2, used=1
6188616704, free=18087424000)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=18182352896, percent=47.0, used=1
6093687808, free=18182352896)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=18127421440, percent=47.1, used=1
6148619264, free=18127421440)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=18130636800, percent=47.1, used=1
6145403904, free=18130636800)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=18281086976, percent=46.7, used=1
5994953728, free=18281086976)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=18274451456, percent=46.7, used=1
6001589248, free=18274451456)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=18270818304, percent=46.7, used=1
6005222400, free=18270818304)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=18207715328, percent=46.9, used=1

6068325376, free=18207715328)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=18204344320, percent=46.9, used=16071696384, free=18204344320)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=18216132608, percent=46.9, used=16059908096, free=18216132608)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=18191319040, percent=46.9, used=16084721664, free=18191319040)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=18196840448, percent=46.9, used=16079200256, free=18196840448)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=18203193344, percent=46.9, used=16072847360, free=18203193344)%
CPU usage: 83.6%
Memory usage: svmem(total=34276040704, available=18185297920, percent=46.9, used=16090742784, free=18185297920)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=18143973376, percent=47.1, used=16132067328, free=18143973376)%
CPU usage: 78.2%
Memory usage: svmem(total=34276040704, available=18134700032, percent=47.1, used=16141340672, free=18134700032)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=18110861312, percent=47.2, used=16165179392, free=18110861312)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=18135703552, percent=47.1, used=16140337152, free=18135703552)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=18122399744, percent=47.1, used=16153640960, free=18122399744)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=18114842624, percent=47.2, used=16161198080, free=18114842624)%
CPU usage: 78.9%
Memory usage: svmem(total=34276040704, available=18127474688, percent=47.1, used=16148566016, free=18127474688)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=18072338432, percent=47.3, used=16203702272, free=18072338432)%
CPU usage: 78.1%
Memory usage: svmem(total=34276040704, available=18088177664, percent=47.2, used=16187863040, free=18088177664)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=18044149760, percent=47.4, used=16231890944, free=18044149760)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=18081812480, percent=47.2, used=16194228224, free=18081812480)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=18049392640, percent=47.3, used=1

6226648064, free=18049392640)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=18017943552, percent=47.4, used=1
6258097152, free=18017943552)%
CPU usage: 80.6%
Memory usage: svmem(total=34276040704, available=18093453312, percent=47.2, used=1
6182587392, free=18093453312)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=18143571968, percent=47.1, used=1
6132468736, free=18143571968)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=18124578816, percent=47.1, used=1
6151461888, free=18124578816)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=18118807552, percent=47.1, used=1
6157233152, free=18118807552)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18122190848, percent=47.1, used=1
6153849856, free=18122190848)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=18077548544, percent=47.3, used=1
6198492160, free=18077548544)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=18040811520, percent=47.4, used=1
6235229184, free=18040811520)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=18016354304, percent=47.4, used=1
6259686400, free=18016354304)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=18031230976, percent=47.4, used=1
6244809728, free=18031230976)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=18015580160, percent=47.4, used=1
6260460544, free=18015580160)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=17998667776, percent=47.5, used=1
6277372928, free=17998667776)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=18016694272, percent=47.4, used=1
6259346432, free=18016694272)%
CPU usage: 82.5%
Memory usage: svmem(total=34276040704, available=18078732288, percent=47.3, used=1
6197308416, free=18078732288)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=18167390208, percent=47.0, used=1
6108650496, free=18167390208)%
CPU usage: 78.3%
Memory usage: svmem(total=34276040704, available=18160967680, percent=47.0, used=1
6115073024, free=18160967680)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=18180050944, percent=47.0, used=1
6095989760, free=18180050944)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=18167627776, percent=47.0, used=1

6108412928, free=18167627776)%
CPU usage: 76.5%
Memory usage: svmem(total=34276040704, available=18191163392, percent=46.9, used=16084877312, free=18191163392)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=18155180032, percent=47.0, used=16120860672, free=18155180032)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=18119831552, percent=47.1, used=16156209152, free=18119831552)%
CPU usage: 74.6%
Memory usage: svmem(total=34276040704, available=18082349056, percent=47.2, used=16193691648, free=18082349056)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18016600064, percent=47.4, used=16259440640, free=18016600064)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=18092003328, percent=47.2, used=16184037376, free=18092003328)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=18027249664, percent=47.4, used=16248791040, free=18027249664)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=18034143232, percent=47.4, used=16241897472, free=18034143232)%
CPU usage: 85.2%
Memory usage: svmem(total=34276040704, available=18122883072, percent=47.1, used=16153157632, free=18122883072)%
CPU usage: 80.3%
Memory usage: svmem(total=34276040704, available=18152497152, percent=47.0, used=16123543552, free=18152497152)%
CPU usage: 80.2%
Memory usage: svmem(total=34276040704, available=18172997632, percent=47.0, used=16103043072, free=18172997632)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=18153967616, percent=47.0, used=16122073088, free=18153967616)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=18155606016, percent=47.0, used=16120434688, free=18155606016)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=18168176640, percent=47.0, used=16107864064, free=18168176640)%
CPU usage: 74.5%
Memory usage: svmem(total=34276040704, available=18188578816, percent=46.9, used=16087461888, free=18188578816)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=18147848192, percent=47.1, used=16128192512, free=18147848192)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=17917853696, percent=47.7, used=16358187008, free=17917853696)%
CPU usage: 97.1%
Memory usage: svmem(total=34276040704, available=17139445760, percent=50.0, used=1

7136594944, free=17139445760)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=17156239360, percent=49.9, used=1
7119801344, free=17156239360)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=17164443648, percent=49.9, used=1
7111597056, free=17164443648)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=17156775936, percent=49.9, used=1
7119264768, free=17156775936)%
CPU usage: 66.4%
Memory usage: svmem(total=34276040704, available=17230331904, percent=49.7, used=1
7045708800, free=17230331904)%
CPU usage: 73.0%
Memory usage: svmem(total=34276040704, available=17108217856, percent=50.1, used=1
7167822848, free=17108217856)%
CPU usage: 88.6%
Memory usage: svmem(total=34276040704, available=17091162112, percent=50.1, used=1
7184878592, free=17091162112)%
CPU usage: 87.4%
Memory usage: svmem(total=34276040704, available=17108152320, percent=50.1, used=1
7167888384, free=17108152320)%
CPU usage: 89.6%
Memory usage: svmem(total=34276040704, available=17087229952, percent=50.1, used=1
7188810752, free=17087229952)%
CPU usage: 88.8%
Memory usage: svmem(total=34276040704, available=17079881728, percent=50.2, used=1
7196158976, free=17079881728)%
CPU usage: 93.3%
Memory usage: svmem(total=34276040704, available=17184919552, percent=49.9, used=1
7091121152, free=17184919552)%
CPU usage: 84.6%
Memory usage: svmem(total=34276040704, available=17219862528, percent=49.8, used=1
7056178176, free=17219862528)%
CPU usage: 87.9%
Memory usage: svmem(total=34276040704, available=17219887104, percent=49.8, used=1
7056153600, free=17219887104)%
CPU usage: 89.1%
Memory usage: svmem(total=34276040704, available=17277833216, percent=49.6, used=1
6998207488, free=17277833216)%
CPU usage: 88.0%
Memory usage: svmem(total=34276040704, available=17208307712, percent=49.8, used=1
7067732992, free=17208307712)%
CPU usage: 84.6%
Memory usage: svmem(total=34276040704, available=17234280448, percent=49.7, used=1
7041760256, free=17234280448)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=17203609600, percent=49.8, used=1
7072431104, free=17203609600)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=17201618944, percent=49.8, used=1
7074421760, free=17201618944)%
CPU usage: 82.6%
Memory usage: svmem(total=34276040704, available=17271832576, percent=49.6, used=1

7004208128, free=17271832576)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17266688000, percent=49.6, used=17009352704, free=17266688000)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17267392512, percent=49.6, used=17008648192, free=17267392512)%
CPU usage: 80.2%
Memory usage: svmem(total=34276040704, available=17262333952, percent=49.6, used=17013706752, free=17262333952)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17246334976, percent=49.7, used=17029705728, free=17246334976)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17254584320, percent=49.7, used=17021456384, free=17254584320)%
CPU usage: 83.2%
Memory usage: svmem(total=34276040704, available=17267433472, percent=49.6, used=17008607232, free=17267433472)%
CPU usage: 85.4%
Memory usage: svmem(total=34276040704, available=17304776704, percent=49.5, used=16971264000, free=17304776704)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=17305141248, percent=49.5, used=16970899456, free=17305141248)%
CPU usage: 80.7%
Memory usage: svmem(total=34276040704, available=17336156160, percent=49.4, used=16939884544, free=17336156160)%
CPU usage: 74.6%
Memory usage: svmem(total=34276040704, available=17264873472, percent=49.6, used=17011167232, free=17264873472)%
CPU usage: 79.1%
Memory usage: svmem(total=34276040704, available=17309921280, percent=49.5, used=16966119424, free=17309921280)%
CPU usage: 74.4%
Memory usage: svmem(total=34276040704, available=17274826752, percent=49.6, used=17001213952, free=17274826752)%
CPU usage: 80.2%
Memory usage: svmem(total=34276040704, available=17299828736, percent=49.5, used=16976211968, free=17299828736)%
CPU usage: 74.1%
Memory usage: svmem(total=34276040704, available=17286746112, percent=49.6, used=16989294592, free=17286746112)%
CPU usage: 74.6%
Memory usage: svmem(total=34276040704, available=17267937280, percent=49.6, used=17008103424, free=17267937280)%
CPU usage: 80.5%
Memory usage: svmem(total=34276040704, available=17256738816, percent=49.7, used=17019301888, free=17256738816)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17265704960, percent=49.6, used=17010335744, free=17265704960)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=17229611008, percent=49.7, used=1

7046429696, free=17229611008)%
CPU usage: 80.1%
Memory usage: svmem(total=34276040704, available=17202622464, percent=49.8, used=17073418240, free=17202622464)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17258332160, percent=49.6, used=17017708544, free=17258332160)%
CPU usage: 79.3%
Memory usage: svmem(total=34276040704, available=17172455424, percent=49.9, used=17103585280, free=17172455424)%
CPU usage: 86.5%
Memory usage: svmem(total=34276040704, available=17231683584, percent=49.7, used=17044357120, free=17231683584)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17272987648, percent=49.6, used=17003053056, free=17272987648)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17284005888, percent=49.6, used=16992034816, free=17284005888)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17282973696, percent=49.6, used=16993067008, free=17282973696)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17252495360, percent=49.7, used=17023545344, free=17252495360)%
CPU usage: 84.4%
Memory usage: svmem(total=34276040704, available=17217626112, percent=49.8, used=17058414592, free=17217626112)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=17286385664, percent=49.6, used=16989655040, free=17286385664)%
CPU usage: 81.9%
Memory usage: svmem(total=34276040704, available=17205850112, percent=49.8, used=17070190592, free=17205850112)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=17245458432, percent=49.7, used=17030582272, free=17245458432)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17240129536, percent=49.7, used=17035911168, free=17240129536)%
CPU usage: 88.9%
Memory usage: svmem(total=34276040704, available=17221808128, percent=49.8, used=17054232576, free=17221808128)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=17174704128, percent=49.9, used=17101336576, free=17174704128)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=17213853696, percent=49.8, used=17062187008, free=17213853696)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=17235062784, percent=49.7, used=17040977920, free=17235062784)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=17219829760, percent=49.8, used=1

7056210944, free=17219829760)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=17213063168, percent=49.8, used=17062977536, free=17213063168)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=17196367872, percent=49.8, used=17079672832, free=17196367872)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17175916544, percent=49.9, used=17100124160, free=17175916544)%
CPU usage: 74.5%
Memory usage: svmem(total=34276040704, available=17163612160, percent=49.9, used=17112428544, free=17163612160)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=17146048512, percent=50.0, used=17129992192, free=17146048512)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=17132937216, percent=50.0, used=17143103488, free=17132937216)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=17120948224, percent=50.0, used=17155092480, free=17120948224)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=17138356224, percent=50.0, used=17137684480, free=17138356224)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=17104523264, percent=50.1, used=17171517440, free=17104523264)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17108537344, percent=50.1, used=17167503360, free=17108537344)%
CPU usage: 80.8%
Memory usage: svmem(total=34276040704, available=17075630080, percent=50.2, used=17200410624, free=17075630080)%
CPU usage: 86.0%
Memory usage: svmem(total=34276040704, available=17184989184, percent=49.9, used=17091051520, free=17184989184)%
CPU usage: 77.5%
Memory usage: svmem(total=34276040704, available=17221603328, percent=49.8, used=17054437376, free=17221603328)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=17305513984, percent=49.5, used=16970526720, free=17305513984)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17284567040, percent=49.6, used=16991473664, free=17284567040)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17260146688, percent=49.6, used=17015894016, free=17260146688)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17253965824, percent=49.7, used=17022074880, free=17253965824)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=17257545728, percent=49.7, used=1

7018494976, free=17257545728)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17246756864, percent=49.7, used=17029283840, free=17246756864)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17222742016, percent=49.8, used=17053298688, free=17222742016)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17197522944, percent=49.8, used=17078517760, free=17197522944)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=17189732352, percent=49.8, used=17086308352, free=17189732352)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=17160736768, percent=49.9, used=17115303936, free=17160736768)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17190445056, percent=49.8, used=17085595648, free=17190445056)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=17186365440, percent=49.9, used=17089675264, free=17186365440)%
CPU usage: 82.8%
Memory usage: svmem(total=34276040704, available=17154269184, percent=50.0, used=17121771520, free=17154269184)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17145249792, percent=50.0, used=17130790912, free=17145249792)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=17145561088, percent=50.0, used=17130479616, free=17145561088)%
CPU usage: 83.1%
Memory usage: svmem(total=34276040704, available=17218842624, percent=49.8, used=17057198080, free=17218842624)%
CPU usage: 78.8%
Memory usage: svmem(total=34276040704, available=17286250496, percent=49.6, used=16989790208, free=17286250496)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=17247211520, percent=49.7, used=17028829184, free=17247211520)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=17248264192, percent=49.7, used=17027776512, free=17248264192)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17235271680, percent=49.7, used=17040769024, free=17235271680)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=17252503552, percent=49.7, used=17023537152, free=17252503552)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=17228820480, percent=49.7, used=17047220224, free=17228820480)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=17183494144, percent=49.9, used=170222742016, free=17183494144)%

7092546560, free=17183494144)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=17155649536, percent=49.9, used=17120391168, free=17155649536)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=17174355968, percent=49.9, used=17101684736, free=17174355968)%
CPU usage: 81.0%
Memory usage: svmem(total=34276040704, available=17145008128, percent=50.0, used=17131032576, free=17145008128)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=17146806272, percent=50.0, used=17129234432, free=17146806272)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=17176756224, percent=49.9, used=17099284480, free=17176756224)%
CPU usage: 86.5%
Memory usage: svmem(total=34276040704, available=17198219264, percent=49.8, used=17077821440, free=17198219264)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=17200746496, percent=49.8, used=17075294208, free=17200746496)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=17194487808, percent=49.8, used=17081552896, free=17194487808)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=17170382848, percent=49.9, used=17105657856, free=17170382848)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=17139019776, percent=50.0, used=17137020928, free=17139019776)%
CPU usage: 80.3%
Memory usage: svmem(total=34276040704, available=17091727360, percent=50.1, used=17184313344, free=17091727360)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=17121869824, percent=50.0, used=17154170880, free=17121869824)%
CPU usage: 80.7%
Memory usage: svmem(total=34276040704, available=17120985088, percent=50.0, used=17155055616, free=17120985088)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=17083916288, percent=50.2, used=17192124416, free=17083916288)%
CPU usage: 81.3%
Memory usage: svmem(total=34276040704, available=17054547968, percent=50.2, used=17221492736, free=17054547968)%
CPU usage: 77.0%
Memory usage: svmem(total=34276040704, available=17080725504, percent=50.2, used=17195315200, free=17080725504)%
CPU usage: 80.6%
Memory usage: svmem(total=34276040704, available=17080311808, percent=50.2, used=17195728896, free=17080311808)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=17039695872, percent=50.3, used=1

7236344832, free=17039695872)%
CPU usage: 80.6%
Memory usage: svmem(total=34276040704, available=17021235200, percent=50.3, used=17254805504, free=17021235200)%
CPU usage: 82.7%
Memory usage: svmem(total=34276040704, available=16982921216, percent=50.5, used=17293119488, free=16982921216)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16926867456, percent=50.6, used=17349173248, free=16926867456)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16925483008, percent=50.6, used=17350557696, free=16925483008)%
CPU usage: 82.4%
Memory usage: svmem(total=34276040704, available=16947257344, percent=50.6, used=17328783360, free=16947257344)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=16966008832, percent=50.5, used=17310031872, free=16966008832)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16951857152, percent=50.5, used=17324183552, free=16951857152)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16980062208, percent=50.5, used=17295978496, free=16980062208)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=16933924864, percent=50.6, used=17342115840, free=16933924864)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=16944177152, percent=50.6, used=17331863552, free=16944177152)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16918056960, percent=50.6, used=17357983744, free=16918056960)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=16927834112, percent=50.6, used=17348206592, free=16927834112)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=16923627520, percent=50.6, used=17352413184, free=16923627520)%
CPU usage: 79.0%
Memory usage: svmem(total=34276040704, available=16919683072, percent=50.6, used=17356357632, free=16919683072)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=16910282752, percent=50.7, used=17365757952, free=16910282752)%
CPU usage: 81.3%
Memory usage: svmem(total=34276040704, available=16906362880, percent=50.7, used=17369677824, free=16906362880)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=16846905344, percent=50.8, used=17429135360, free=16846905344)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=16811646976, percent=51.0, used=1

7464393728, free=16811646976)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=16881782784, percent=50.7, used=17394257920, free=16881782784)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16911728640, percent=50.7, used=17364312064, free=16911728640)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16915476480, percent=50.6, used=17360564224, free=16915476480)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16879669248, percent=50.8, used=17396371456, free=16879669248)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=16882233344, percent=50.7, used=17393807360, free=16882233344)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=16881897472, percent=50.7, used=17394143232, free=16881897472)%
CPU usage: 74.5%
Memory usage: svmem(total=34276040704, available=16822968320, percent=50.9, used=17453072384, free=16822968320)%
CPU usage: 80.9%
Memory usage: svmem(total=34276040704, available=16890597376, percent=50.7, used=17385443328, free=16890597376)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16888307712, percent=50.7, used=17387732992, free=16888307712)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=16889040896, percent=50.7, used=17386999808, free=16889040896)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=16845910016, percent=50.9, used=17430130688, free=16845910016)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=16874651648, percent=50.8, used=17401389056, free=16874651648)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16838680576, percent=50.9, used=17437360128, free=16838680576)%
CPU usage: 75.1%
Memory usage: svmem(total=34276040704, available=16853295104, percent=50.8, used=17422745600, free=16853295104)%
CPU usage: 80.2%
Memory usage: svmem(total=34276040704, available=16923480064, percent=50.6, used=17352560640, free=16923480064)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=16968364032, percent=50.5, used=17307676672, free=16968364032)%
CPU usage: 74.0%
Memory usage: svmem(total=34276040704, available=16951017472, percent=50.5, used=17325023232, free=16951017472)%
CPU usage: 77.4%
Memory usage: svmem(total=34276040704, available=16948183040, percent=50.6, used=1

7327857664, free=16948183040)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=17031700480, percent=50.3, used=17244340224, free=17031700480)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17018597376, percent=50.3, used=17257443328, free=17018597376)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17021444096, percent=50.3, used=17254596608, free=17021444096)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=17024495616, percent=50.3, used=17251545088, free=17024495616)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=17040728064, percent=50.3, used=17235312640, free=17040728064)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17047752704, percent=50.3, used=17228288000, free=17047752704)%
CPU usage: 88.0%
Memory usage: svmem(total=34276040704, available=17178619904, percent=49.9, used=17097420800, free=17178619904)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=17082081280, percent=50.2, used=17193959424, free=17082081280)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=17129422848, percent=50.0, used=17146617856, free=17129422848)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=17119322112, percent=50.1, used=17156718592, free=17119322112)%
CPU usage: 84.5%
Memory usage: svmem(total=34276040704, available=17035841536, percent=50.3, used=17240199168, free=17035841536)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=17119412224, percent=50.1, used=17156628480, free=17119412224)%
CPU usage: 79.8%
Memory usage: svmem(total=34276040704, available=17100693504, percent=50.1, used=17175347200, free=17100693504)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=17058471936, percent=50.2, used=17217568768, free=17058471936)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=17086218240, percent=50.2, used=17189822464, free=17086218240)%
CPU usage: 76.8%
Memory usage: svmem(total=34276040704, available=17095610368, percent=50.1, used=17180430336, free=17095610368)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=17119977472, percent=50.1, used=17156063232, free=17119977472)%
CPU usage: 79.3%
Memory usage: svmem(total=34276040704, available=17118224384, percent=50.1, used=1

7157816320, free=17118224384)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17125482496, percent=50.0, used=17150558208, free=17125482496)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=17115119616, percent=50.1, used=17160921088, free=17115119616)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=17102204928, percent=50.1, used=1713835776, free=17102204928)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17121398784, percent=50.0, used=17154641920, free=17121398784)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=17055322112, percent=50.2, used=17220718592, free=17055322112)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=17061310464, percent=50.2, used=17214730240, free=17061310464)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=17104531456, percent=50.1, used=17171509248, free=17104531456)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=17048891392, percent=50.3, used=17227149312, free=17048891392)%
CPU usage: 74.4%
Memory usage: svmem(total=34276040704, available=17036894208, percent=50.3, used=17239146496, free=17036894208)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=17015640064, percent=50.4, used=17260400640, free=17015640064)%
CPU usage: 74.8%
Memory usage: svmem(total=34276040704, available=17017729024, percent=50.4, used=17258311680, free=17017729024)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=17049587712, percent=50.3, used=17226452992, free=17049587712)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16940920832, percent=50.6, used=17335119872, free=16940920832)%
CPU usage: 88.9%
Memory usage: svmem(total=34276040704, available=16904740864, percent=50.7, used=17371299840, free=16904740864)%
CPU usage: 74.5%
Memory usage: svmem(total=34276040704, available=16904773632, percent=50.7, used=17371267072, free=16904773632)%
CPU usage: 74.4%
Memory usage: svmem(total=34276040704, available=16853204992, percent=50.8, used=17422835712, free=16853204992)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16868814848, percent=50.8, used=17407225856, free=16868814848)%
CPU usage: 75.4%
Memory usage: svmem(total=34276040704, available=16848633856, percent=50.8, used=1

7427406848, free=16848633856)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=16853905408, percent=50.8, used=17422135296, free=16853905408)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=16830693376, percent=50.9, used=17445347328, free=16830693376)%
CPU usage: 79.6%
Memory usage: svmem(total=34276040704, available=16768897024, percent=51.1, used=17507143680, free=16768897024)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=16763731968, percent=51.1, used=17512308736, free=16763731968)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=16798441472, percent=51.0, used=17477599232, free=16798441472)%
CPU usage: 81.4%
Memory usage: svmem(total=34276040704, available=16829792256, percent=50.9, used=17446248448, free=16829792256)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=16875012096, percent=50.8, used=17401028608, free=16875012096)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=16882225152, percent=50.7, used=17393815552, free=16882225152)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16863637504, percent=50.8, used=17412403200, free=16863637504)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16864018432, percent=50.8, used=17412022272, free=16864018432)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=16826216448, percent=50.9, used=17449824256, free=16826216448)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16817692672, percent=50.9, used=17458348032, free=16817692672)%
CPU usage: 81.3%
Memory usage: svmem(total=34276040704, available=16817684480, percent=50.9, used=17458356224, free=16817684480)%
CPU usage: 85.1%
Memory usage: svmem(total=34276040704, available=16812642304, percent=50.9, used=17463398400, free=16812642304)%
CPU usage: 78.9%
Memory usage: svmem(total=34276040704, available=16780922880, percent=51.0, used=17495117824, free=16780922880)%
CPU usage: 78.5%
Memory usage: svmem(total=34276040704, available=16780693504, percent=51.0, used=17495347200, free=16780693504)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=16792322048, percent=51.0, used=17483718656, free=16792322048)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16783122432, percent=51.0, used=1

7492918272, free=16783122432)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=16763887616, percent=51.1, used=17512153088, free=16763887616)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=16808820736, percent=51.0, used=17467219968, free=16808820736)%
CPU usage: 84.7%
Memory usage: svmem(total=34276040704, available=16686501888, percent=51.3, used=17589538816, free=16686501888)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=16741822464, percent=51.2, used=17534218240, free=16741822464)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=16726880256, percent=51.2, used=17549160448, free=16726880256)%
CPU usage: 82.9%
Memory usage: svmem(total=34276040704, available=16713396224, percent=51.2, used=17562644480, free=16713396224)%
CPU usage: 79.9%
Memory usage: svmem(total=34276040704, available=16739041280, percent=51.2, used=17536999424, free=16739041280)%
CPU usage: 84.2%
Memory usage: svmem(total=34276040704, available=16784793600, percent=51.0, used=17491247104, free=16784793600)%
CPU usage: 82.3%
Memory usage: svmem(total=34276040704, available=16830464000, percent=50.9, used=17445576704, free=16830464000)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16829460480, percent=50.9, used=17446580224, free=16829460480)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16821178368, percent=50.9, used=17454862336, free=16821178368)%
CPU usage: 82.8%
Memory usage: svmem(total=34276040704, available=16829607936, percent=50.9, used=17446432768, free=16829607936)%
CPU usage: 81.8%
Memory usage: svmem(total=34276040704, available=16825896960, percent=50.9, used=17450143744, free=16825896960)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=16806068224, percent=51.0, used=17469972480, free=16806068224)%
CPU usage: 79.3%
Memory usage: svmem(total=34276040704, available=16759791616, percent=51.1, used=17516249088, free=16759791616)%
CPU usage: 79.1%
Memory usage: svmem(total=34276040704, available=16773480448, percent=51.1, used=17502560256, free=16773480448)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=16786370560, percent=51.0, used=17489670144, free=16786370560)%
CPU usage: 77.2%
Memory usage: svmem(total=34276040704, available=16749338624, percent=51.1, used=1

7526702080, free=16749338624)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=16785776640, percent=51.0, used=17490264064, free=16785776640)%
CPU usage: 83.9%
Memory usage: svmem(total=34276040704, available=16728137728, percent=51.2, used=17547902976, free=16728137728)%
CPU usage: 81.6%
Memory usage: svmem(total=34276040704, available=16784494592, percent=51.0, used=17491546112, free=16784494592)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16779616256, percent=51.0, used=17496424448, free=16779616256)%
CPU usage: 75.5%
Memory usage: svmem(total=34276040704, available=16753217536, percent=51.1, used=17522823168, free=16753217536)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=16787722240, percent=51.0, used=17488318464, free=16787722240)%
CPU usage: 77.6%
Memory usage: svmem(total=34276040704, available=16774582272, percent=51.1, used=17501458432, free=16774582272)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=16822968320, percent=50.9, used=17453072384, free=16822968320)%
CPU usage: 82.2%
Memory usage: svmem(total=34276040704, available=16731803648, percent=51.2, used=17544237056, free=16731803648)%
CPU usage: 81.0%
Memory usage: svmem(total=34276040704, available=16755757056, percent=51.1, used=17520283648, free=16755757056)%
CPU usage: 84.6%
Memory usage: svmem(total=34276040704, available=16849080320, percent=50.8, used=17426960384, free=16849080320)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=16871084032, percent=50.8, used=17404956672, free=16871084032)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16827383808, percent=50.9, used=17448656896, free=16827383808)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=16836218880, percent=50.9, used=17439821824, free=16836218880)%
CPU usage: 83.9%
Memory usage: svmem(total=34276040704, available=16855523328, percent=50.8, used=17420517376, free=16855523328)%
CPU usage: 81.5%
Memory usage: svmem(total=34276040704, available=16859529216, percent=50.8, used=17416511488, free=16859529216)%
CPU usage: 75.6%
Memory usage: svmem(total=34276040704, available=16856326144, percent=50.8, used=17419714560, free=16856326144)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=16857812992, percent=50.8, used=1

7418227712, free=16857812992)%
CPU usage: 77.4%
Memory usage: svmem(total=34276040704, available=16840687616, percent=50.9, used=17435353088, free=16840687616)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=16829149184, percent=50.9, used=17446891520, free=16829149184)%
CPU usage: 76.0%
Memory usage: svmem(total=34276040704, available=16838033408, percent=50.9, used=17438007296, free=16838033408)%
CPU usage: 75.3%
Memory usage: svmem(total=34276040704, available=16814809088, percent=50.9, used=17461231616, free=16814809088)%
CPU usage: 76.9%
Memory usage: svmem(total=34276040704, available=16872017920, percent=50.8, used=17404022784, free=16872017920)%
CPU usage: 83.0%
Memory usage: svmem(total=34276040704, available=16892604416, percent=50.7, used=17383436288, free=16892604416)%
CPU usage: 78.9%
Memory usage: svmem(total=34276040704, available=16898514944, percent=50.7, used=17377525760, free=16898514944)%
CPU usage: 79.5%
Memory usage: svmem(total=34276040704, available=16924401664, percent=50.6, used=17351639040, free=16924401664)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=16907784192, percent=50.7, used=17368256512, free=16907784192)%
CPU usage: 80.9%
Memory usage: svmem(total=34276040704, available=16904695808, percent=50.7, used=17371344896, free=16904695808)%
CPU usage: 80.7%
Memory usage: svmem(total=34276040704, available=16922591232, percent=50.6, used=17353449472, free=16922591232)%
CPU usage: 80.1%
Memory usage: svmem(total=34276040704, available=16864526336, percent=50.8, used=17411514368, free=16864526336)%
CPU usage: 73.8%
Memory usage: svmem(total=34276040704, available=16857534464, percent=50.8, used=17418506240, free=16857534464)%
CPU usage: 77.0%
Memory usage: svmem(total=34276040704, available=16829489152, percent=50.9, used=17446551552, free=16829489152)%
CPU usage: 79.7%
Memory usage: svmem(total=34276040704, available=16841732096, percent=50.9, used=17434308608, free=16841732096)%
CPU usage: 76.6%
Memory usage: svmem(total=34276040704, available=16845541376, percent=50.9, used=17430499328, free=16845541376)%
CPU usage: 76.7%
Memory usage: svmem(total=34276040704, available=16888979456, percent=50.7, used=17387061248, free=16888979456)%
CPU usage: 80.2%
Memory usage: svmem(total=34276040704, available=16854605824, percent=50.8, used=1

7421434880, free=16854605824)%
CPU usage: 77.3%
Memory usage: svmem(total=34276040704, available=16848732160, percent=50.8, used=17427308544, free=16848732160)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16846282752, percent=50.9, used=17429757952, free=16846282752)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=16831516672, percent=50.9, used=17444524032, free=16831516672)%
CPU usage: 80.1%
Memory usage: svmem(total=34276040704, available=16840605696, percent=50.9, used=17435435008, free=16840605696)%
CPU usage: 81.5%
Memory usage: svmem(total=34276040704, available=16816074752, percent=50.9, used=17459965952, free=16816074752)%
CPU usage: 81.7%
Memory usage: svmem(total=34276040704, available=16899993600, percent=50.7, used=17376047104, free=16899993600)%
CPU usage: 84.9%
Memory usage: svmem(total=34276040704, available=16860864512, percent=50.8, used=17415176192, free=16860864512)%
CPU usage: 82.8%
Memory usage: svmem(total=34276040704, available=16889286656, percent=50.7, used=17386754048, free=16889286656)%
CPU usage: 76.1%
Memory usage: svmem(total=34276040704, available=16886824960, percent=50.7, used=17389215744, free=16886824960)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=16944173056, percent=50.6, used=17331867648, free=16944173056)%
CPU usage: 75.2%
Memory usage: svmem(total=34276040704, available=16903495680, percent=50.7, used=17372545024, free=16903495680)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=16900665344, percent=50.7, used=17375375360, free=16900665344)%
CPU usage: 79.3%
Memory usage: svmem(total=34276040704, available=16906387456, percent=50.7, used=17369653248, free=16906387456)%
CPU usage: 85.6%
Memory usage: svmem(total=34276040704, available=16847769600, percent=50.8, used=17428271104, free=16847769600)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=16864546816, percent=50.8, used=17411493888, free=16864546816)%
CPU usage: 76.3%
Memory usage: svmem(total=34276040704, available=16887017472, percent=50.7, used=17389023232, free=16887017472)%
CPU usage: 82.6%
Memory usage: svmem(total=34276040704, available=16878551040, percent=50.8, used=17397489664, free=16878551040)%
CPU usage: 80.0%
Memory usage: svmem(total=34276040704, available=16873684992, percent=50.8, used=1

7402355712, free=16873684992)%
CPU usage: 75.0%
Memory usage: svmem(total=34276040704, available=16851668992, percent=50.8, used=1
7424371712, free=16851668992)%
CPU usage: 84.4%
Memory usage: svmem(total=34276040704, available=16831070208, percent=50.9, used=1
7444970496, free=16831070208)%
CPU usage: 75.8%
Memory usage: svmem(total=34276040704, available=16853790720, percent=50.8, used=1
7422249984, free=16853790720)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16916660224, percent=50.6, used=1
7359380480, free=16916660224)%
CPU usage: 84.4%
Memory usage: svmem(total=34276040704, available=16761974784, percent=51.1, used=1
7514065920, free=16761974784)%
CPU usage: 77.7%
Memory usage: svmem(total=34276040704, available=16746459136, percent=51.1, used=1
7529581568, free=16746459136)%
CPU usage: 77.9%
Memory usage: svmem(total=34276040704, available=16853143552, percent=50.8, used=1
7422897152, free=16853143552)%
CPU usage: 81.2%
Memory usage: svmem(total=34276040704, available=16815616000, percent=50.9, used=1
7460424704, free=16815616000)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=16823148544, percent=50.9, used=1
7452892160, free=16823148544)%
CPU usage: 77.1%
Memory usage: svmem(total=34276040704, available=16841506816, percent=50.9, used=1
7434533888, free=16841506816)%
CPU usage: 77.4%
Memory usage: svmem(total=34276040704, available=16821673984, percent=50.9, used=1
7454366720, free=16821673984)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16831098880, percent=50.9, used=1
7444941824, free=16831098880)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16819355648, percent=50.9, used=1
7456685056, free=16819355648)%
CPU usage: 75.9%
Memory usage: svmem(total=34276040704, available=16841916416, percent=50.9, used=1
7434124288, free=16841916416)%
CPU usage: 74.7%
Memory usage: svmem(total=34276040704, available=16808058880, percent=51.0, used=1
7467981824, free=16808058880)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16828534784, percent=50.9, used=1
7447505920, free=16828534784)%
CPU usage: 76.4%
Memory usage: svmem(total=34276040704, available=16789532672, percent=51.0, used=1
7486508032, free=16789532672)%
CPU usage: 78.0%
Memory usage: svmem(total=34276040704, available=16831696896, percent=50.9, used=1

7444343808, free=16831696896)%
CPU usage: 83.7%
Memory usage: svmem(total=34276040704, available=16810401792, percent=51.0, used=17465638912, free=16810401792)%
CPU usage: 76.2%
Memory usage: svmem(total=34276040704, available=16807174144, percent=51.0, used=17468866560, free=16807174144)%
CPU usage: 83.1%
Memory usage: svmem(total=34276040704, available=16848809984, percent=50.8, used=17427230720, free=16848809984)%
CPU usage: 78.4%
Memory usage: svmem(total=34276040704, available=16865271808, percent=50.8, used=17410768896, free=16865271808)%
CPU usage: 74.9%
Memory usage: svmem(total=34276040704, available=16848236544, percent=50.8, used=17427804160, free=16848236544)%
CPU usage: 78.6%
Memory usage: svmem(total=34276040704, available=16903430144, percent=50.7, used=17372610560, free=16903430144)%
CPU usage: 76.7%
Memory usage: svmem(total=34276040704, available=16893579264, percent=50.7, used=17382461440, free=16893579264)%
CPU usage: 75.7%
Memory usage: svmem(total=34276040704, available=16884109312, percent=50.7, used=17391931392, free=16884109312)%
CPU usage: 76.5%
Memory usage: svmem(total=34276040704, available=16865091584, percent=50.8, used=17410949120, free=16865091584)%
CPU usage: 55.4%
Memory usage: svmem(total=34276040704, available=16853557248, percent=50.8, used=17422483456, free=16853557248)%
CPU usage: 49.5%
Memory usage: svmem(total=34276040704, available=16850706432, percent=50.8, used=17425334272, free=16850706432)%
CPU usage: 43.9%
Memory usage: svmem(total=34276040704, available=17032843264, percent=50.3, used=17243197440, free=17032843264)%

Evaluation Metrics:

	Train Set	Test Set
MSE	178.409506	365.090424
RMSE	13.357002	19.107340
MAE	5.696037	8.364257
R2	0.998070	0.996064
EVS	0.998070	0.996065
MAPE	0.010674	0.015629

Parameters

regressor__colsample_bytree	0.8
regressor__gamma	0.25
regressor__learning_rate	0.04
regressor__max_depth	65
regressor__min_child_weight	16
regressor__n_estimators	600
regressor__reg_alpha	40

```

regressor__reg_lambda          0
regressor__subsample           0.9
regressor__tree_method         gpu_hist
Computation time: 5299.637978792191

```

Evaluation Metrics:

	XGB Train	XGB Test
MSE	178.409506	365.090424
RMSE	13.357002	19.107340
MAE	5.696037	8.364257
R2	0.998070	0.996064
EVS	0.998070	0.996065
MAPE	0.010674	0.015629

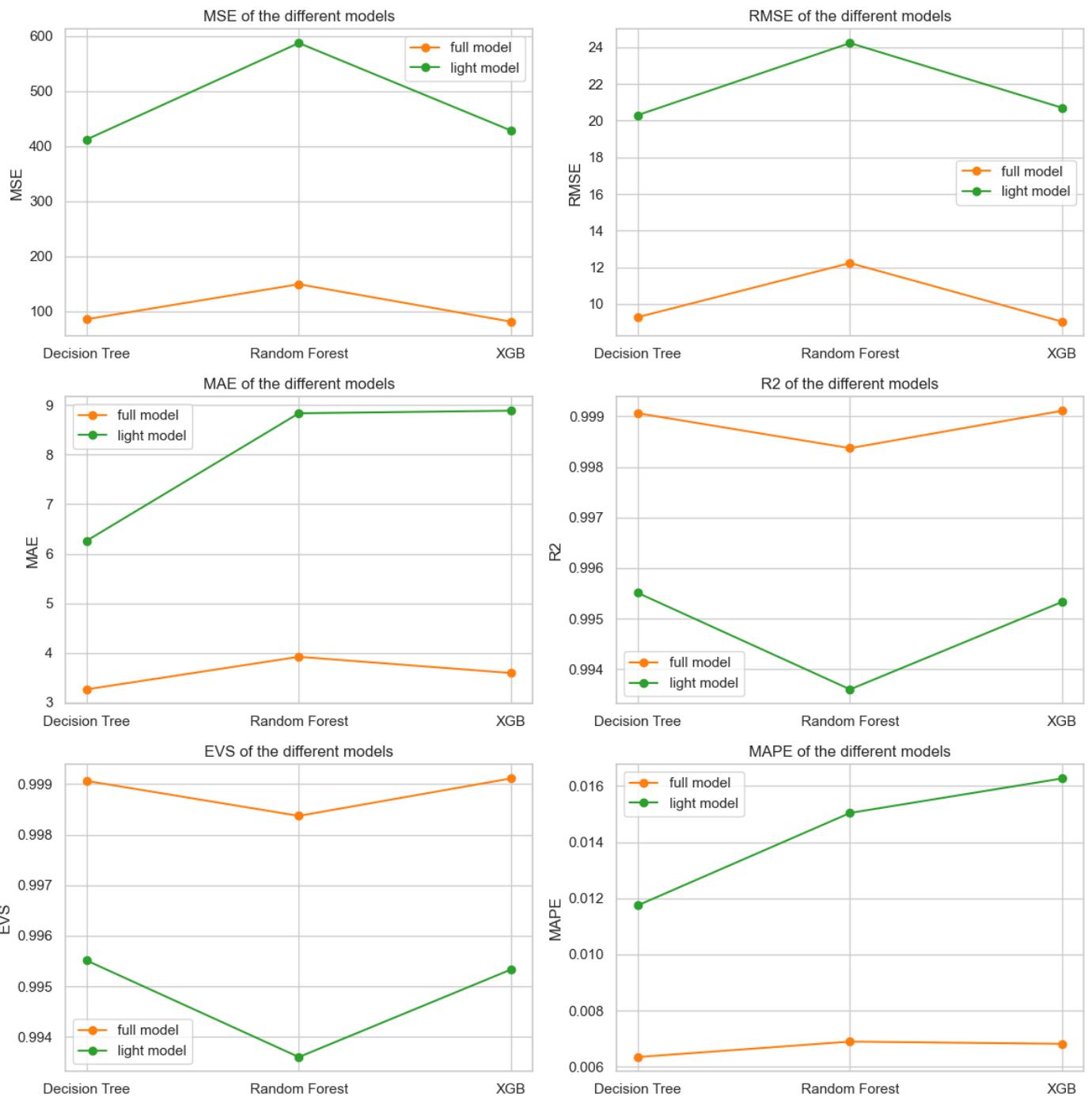
A6.2 Performance full vs light models

Decision Tree out of sample Random Forest out of sample XGB out of sample

MSE	412.294540	587.562125	428.523895
RMSE	20.305037	24.239681	20.700819
MAE	6.260306	8.837030	8.888611
R2	0.995510	0.993601	0.995333
EVS	0.995512	0.993601	0.995334
MAPE	0.011746	0.015039	0.016270

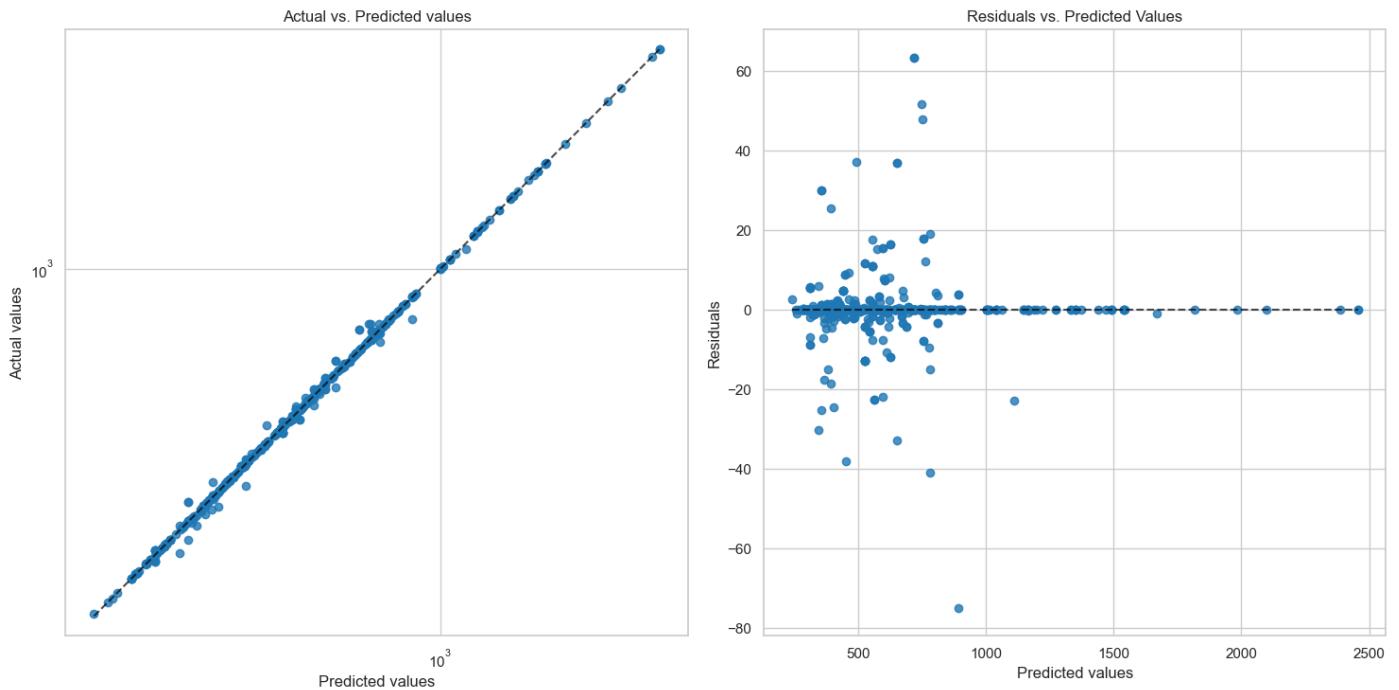
Decision Tree out of sample Random Forest out of sample XGB out of sample

MSE	86.005198	149.547716	81.498908
RMSE	9.273899	12.228970	9.027675
MAE	3.260411	3.918312	3.590986
R2	0.999063	0.998371	0.999112
EVS	0.999064	0.998371	0.999113
MAPE	0.006344	0.006896	0.006816



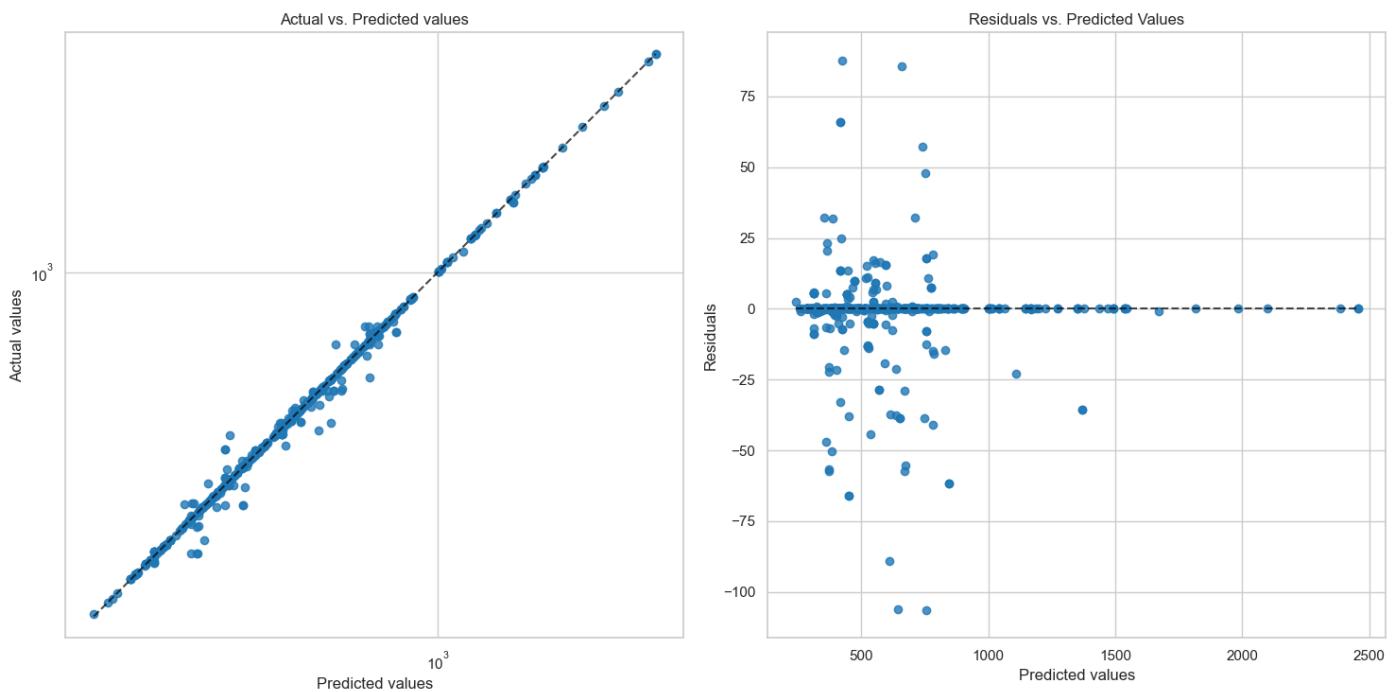
A6.2.1 Performance full vs light models

Plotting cross-validated predictions of Decision Tree



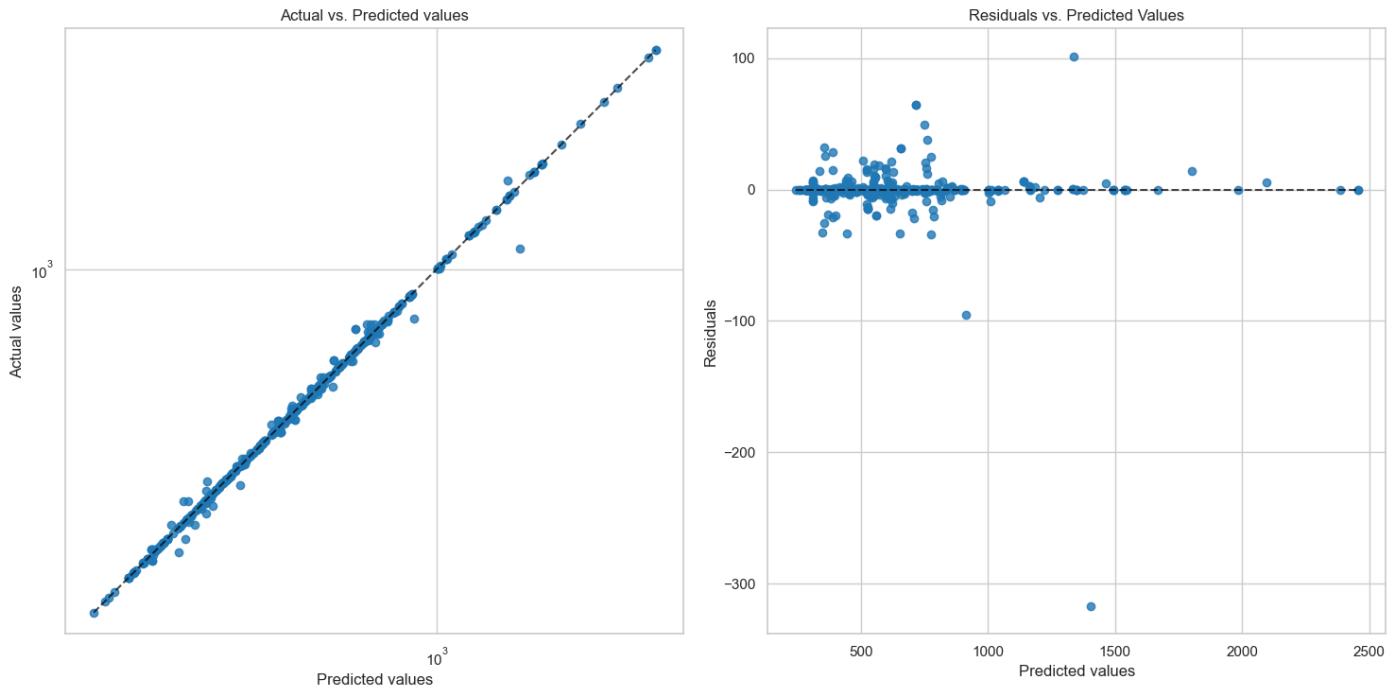
The mean residual of Decision Tree is: -0.14192113936983314

Plotting cross-validated predictions of Decision Tree light



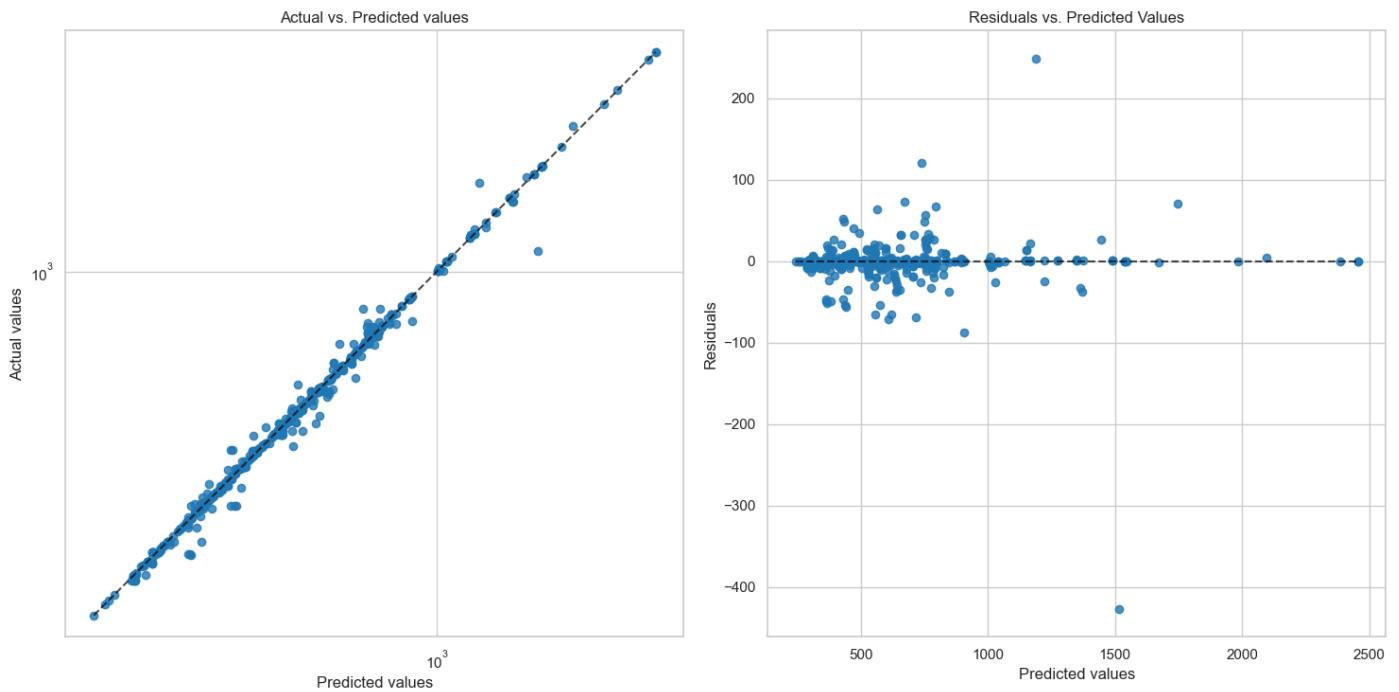
The mean residual of Decision Tree light is: -0.46648864660616046

Plotting cross-validated predictions of Random Forest



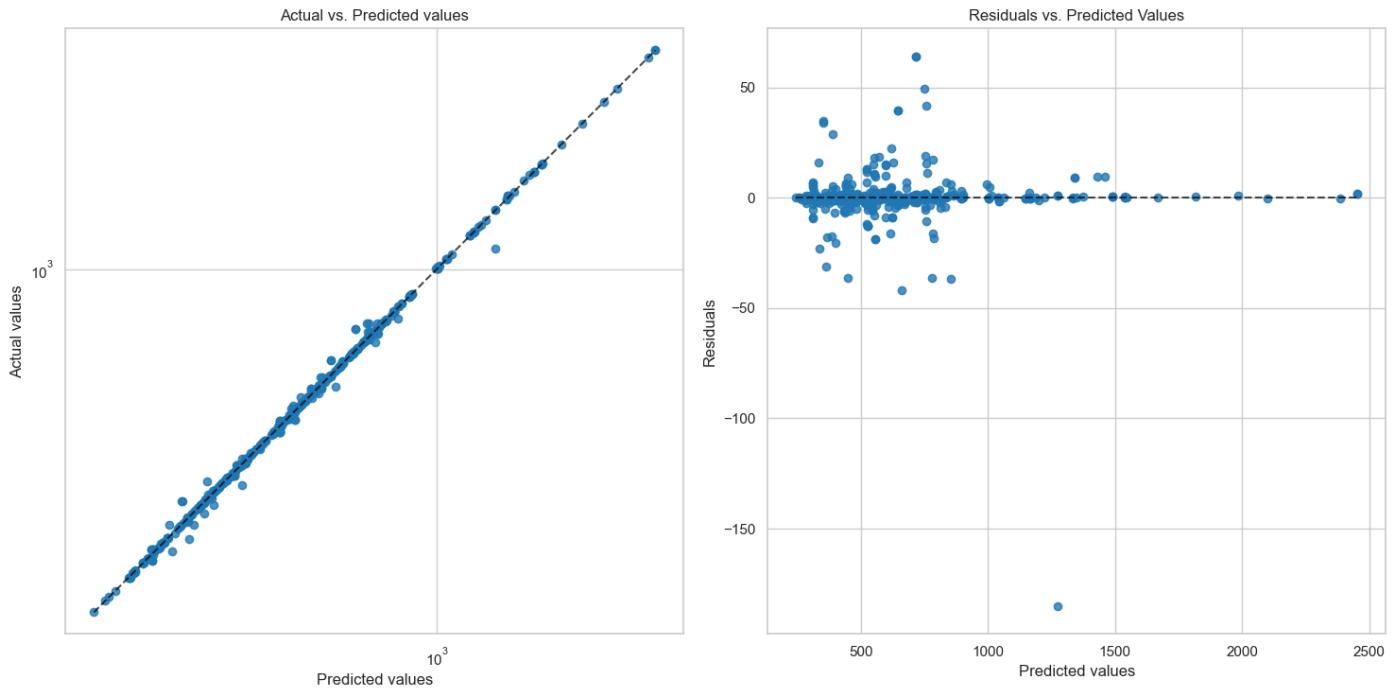
The mean residual of Random Forest is: -0.0004747767460958373

Plotting cross-validated predictions of Random Forest light



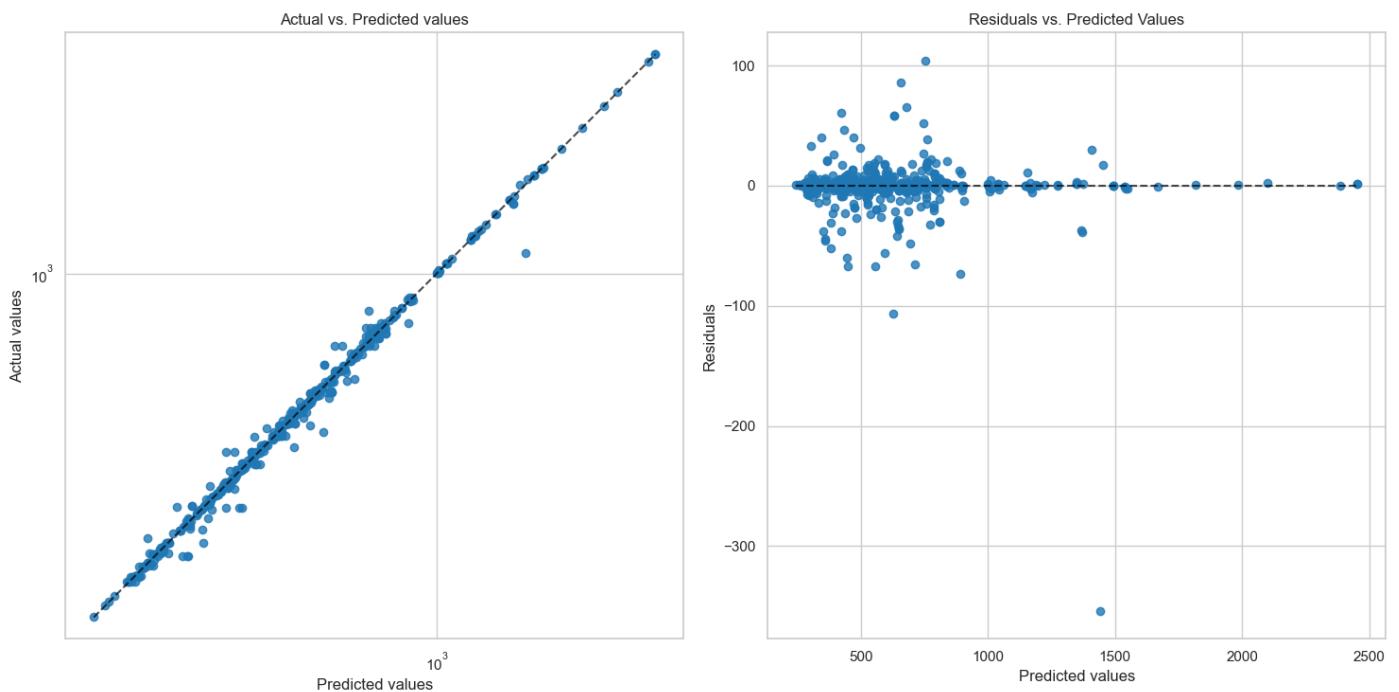
The mean residual of Random Forest light is: 0.10922425080982177

Plotting cross-validated predictions of XGB



The mean residual of XGB is: -0.10674510261257672

Plotting cross-validated predictions of XGB light



The mean residual of XGB light is: -0.3072127171791553

Still open Problems to be solved

References!

GUI

- Implement reset button
- Hidden advanced options should be deleted
- Registration in Date, not month
- KW/PS choosable
- Cat vars not prefilled