Data Mining Project
Meta Dataset Regression

Marco Sau
60/79/00028

February 2024

Contents

1 Introduction

1.1 Problem Description

This project aims to conduct an in-depth analysis of the "Meta" dataset, a set of data in which each object represents the characteristics of a dataset of literature on which a learning algorithm is applied. The project aims to showcase the application of various regression models and data analysis methods to draw conclusions and make predictions based on the dataset's attributes.

1.2 Objectives

The primary objective of this project is:

- Apply regression techniques.

Related objectives are:

- Perform an in-depth analysis of the Meta dataset.
- Draw meaningful insights and conclusions from the analysis.

2 Dataset

2.1 Description

The dataset is publicly available on OpenML.
The original name of the dataset is Meta-Data, and it was used to advise about which classification method is appropriate for a particular dataset.
This Data is about the results of Statlog project. The project performed a comparative study between Statistical, Neural, and Symbolic learning algorithms.
The dataset is created for classification tasks, not for regression tasks.
Conducting regression analysis on a dataset originally intended for classification might necessitate modifications to both the dataset and the methodologies involved.

2.2 Main Features

The list of attributes is described in Table 1:

Table 1: Dataset Attributes Description

No.	Attribute	Type	Description
1	DS_Name	categorical	Name of DataSet
2	T	continuous	Number of examples in test set
3	N	continuous	Number of examples
4	p	continuous	Number of attributes
5	k	continuous	Number of classes
6	Bin	continuous	Number of binary Attributes
7	Cost	continuous	Cost (1=yes,0=no)
8	SDRatio	continuous	Standard deviation ratio
9	correl	continuous	Mean correlation between attributes
10	cancor1	continuous	First canonical correlation
11	cancor2	continuous	Second canonical correlation
12	fract1	continuous	First eigenvalue
13	fract2	continuous	Second eigenvalue
14	skewness	continuous	Mean of $\frac{\lvert E[(X-Mean)\rvert^3}{STD^3}$
15	kurtosis	continuous	Mean of $\frac{E[(X-Mean)^4]}{STD^4}$
16	Hc	continuous	Mean entropy of attributes
17	Hx	continuous	Entropy of classes
18	MCx	continuous	Mean mutual entropy of class and attributes
19	EnAtr	continuous	Equivalent number of attributes
20	NSRatio	continuous	Noise-signal ratio
21	Alg_Name	categorical	Name of Algorithm
22	Norm_error	continuous	Normalized Error (continuous class)

Key characteristics of the dataset are:

- 22 attributes:
  - 20 numerical;
  - 2 categorical, DS_Name and Alg_name.
- 528 instances
- The target variable, has 436 values.
- 504 missing values distributed as follows:

  - correl: 24 missing values
  - cancor2: 240 missing values
  - fract2: 240 missing values
- Based on the documentation, fract2 and cancor 2 only apply to datasets with more than 2 classes.
- 264 instances with missing values
- 55 duplicate target values
- 381 instances with a unique target value
- 147 instances that have the same target value as another

The distribution of the attributes' values is shown in Figure 1.

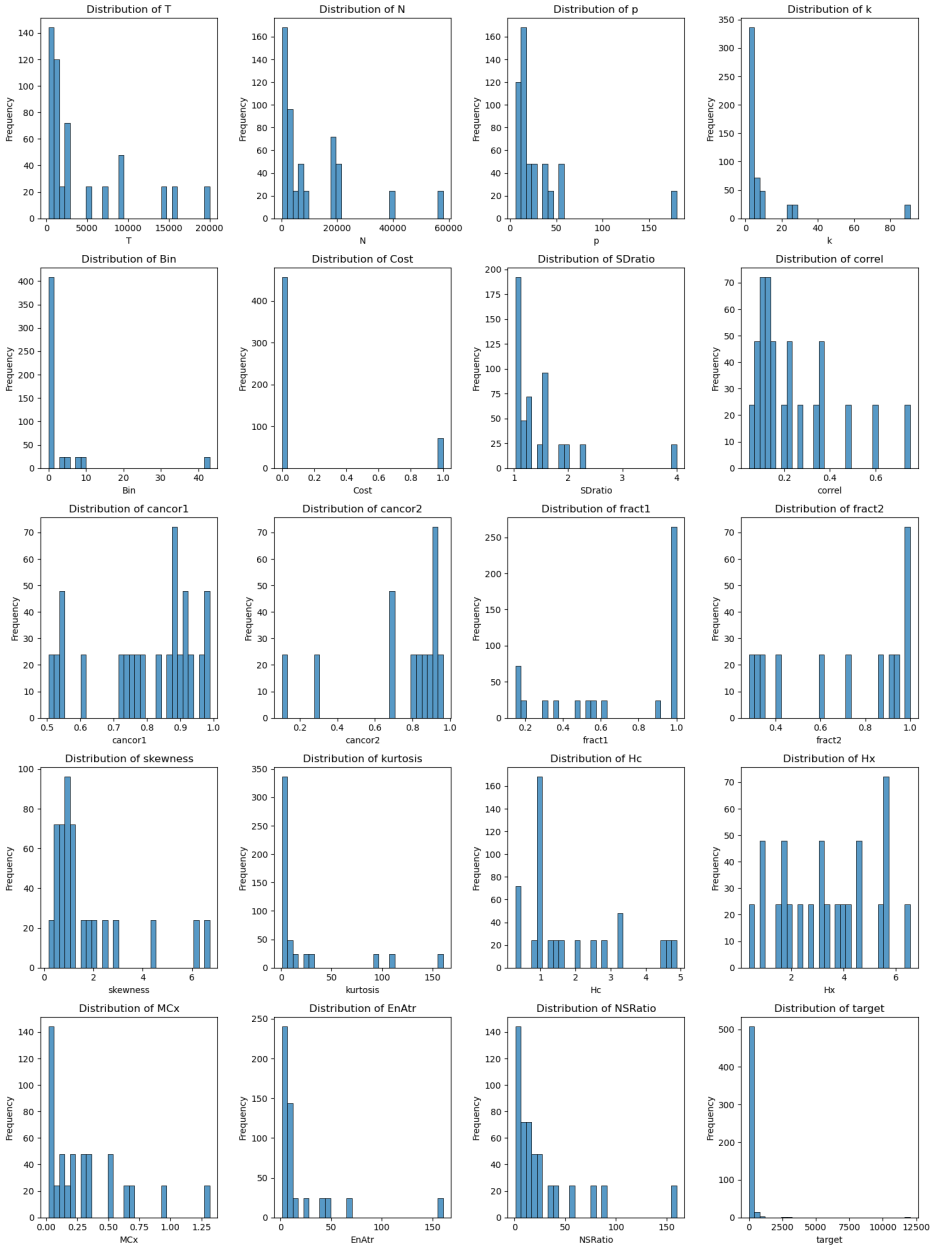

Figure 1: Numeric Distribution

Observations:

- The target variable has a large range of values, mostly condensed around zero. The extreme values could be outliers.
- Cost assumes only two values, 0 and 1.
- p, k, SDratio, and Bin have a high gap between the higher values and the rest of the numbers.
- 10 variables have most values around zero.
- fract1 and fract2 have most of the values equal to 1.

3 Pre-Processing

Two data elaboration techniques and two transformation techniques have been applied in pre-processing. The elaboration techniques concern the elimination of attributes, therefore of columns, and the elimination of rows, therefore of instances, while the transformation techniques concern the substitution of missing values with values appropriately generated by a specific function, and the standardization of the data before the training step.

3.1 Preliminary Steps

The dataset has been downloaded using fetch_openml(name='meta', version=1) and organized in a pandas dataframe. Additionally, a physical version of the dataset has been provided.

3.2 Removing Categorical Attributes

During this process, we removed the nominal features ('DS_Name' and 'Alg_Name') as not relevant to our regression tasks.

3.3 Handling Missing Values

We managed the missing values in two different ways and created two separate datasets: D1 and D2.

3.3.1 Finding Missing Values

First, using pandas, we searched for incomplete information in the data, and the results are summarized in Table 2:

Column	Count
T	0
N	0
p	0
k	0
Bin	0
Cost	0
SDratio	0
correl	24
cancor1	0
cancor2	240
fract1	0
fract2	240
skewness	0
kurtosis	0
Hc	0
Hx	0
MCx	0
ENAt	0
NSRatio	0
target	0

Table 2: Counts of Non-Null Values per Column

Observations:
We have found 3 attributes with missing values, correl, cancor2, and fract2.

3.3.2 D1: removing rows with missing values

One of the research questions is to create a dataset without the instances with missing values, which has been called D1.
We have identified the rows with missing values, and then we have deleted them.
This approach, while simple, can lead to the loss of important information, especially if the missing data is numerous. Generally speaking, removing a high number of rows is a bad practice and should be avoided.
The new dataset now has 264 rows, which is exactly half of the size of the original dataset.
A summary of dataset D1's attributes is shown in Table 3.

#	Attribute	Non-Null Count
1	T	264 non-null
2	N	264 non-null
3	p	264 non-null
4	k	264 non-null
5	Bin	264 non-null
6	Cost	264 non-null
7	SDratio	264 non-null
8	correl	264 non-null
9	cancor1	264 non-null
10	cancor2	264 non-null
11	fract1	264 non-null
12	fract2	264 non-null
13	skewness	264 non-null
14	kurtosis	264 non-null
15	Hc	264 non-null
16	Hx	264 non-null
17	MCx	264 non-null
18	ENAtr	264 non-null
19	NSRatio	264 non-null
20	target	264 non-null

Table 3: Summary of Dataset D1 Attributes

An investigation has been made to understand the nature of the target variable in D1, main results are:

- 22 duplicate target values
- 209 instances with a unique target value
- 55 instances that have the same target value as another

3.3.3 D2: interpolate missing values

Another research question is to create a new dataset where the missing values must be replaced by using interpolation with the existing data. The dataset's name is D2.
We used the 'interpolated' pandas function to interpolate missing data, choosing the 'cubicspline' method. This approach attempts to estimate missing values based on existing data, thus retaining more information than simple deletion. The main parameters we have used are the following:

- 'method='cubicspline'': This specifies the interpolation technique to be used. A cubic spline is a smooth, piecewise-defined function built from cubic polynomials, which is used to approximate the underlying function that the data points might represent. It ensures that the interpolated values will form a smooth curve that fits closely to the actual data points.

- 'limit_direction='both'': This parameter determines the direction in which to apply the interpolation. When set to 'both', it means that pandas will fill missing values in both forward and backward directions (i.e., it will not only fill NaNs using the values that come before them but also using the values that come after them).
- 'axis=0': This indicates that the interpolation should be applied vertically, column by column.

When this method is applied to a DataFrame, pandas will go through each column and replace missing values with the result of the cubic spline interpolation, based on the non-missing values in that column. This type of interpolation is often used when the data is believed to follow a non-linear pattern, and you require a smooth approximation of the missing values rather than a linear one.
A summary of dataset D2's attributes is shown in Table 4.

#	**Column**	**Non-Null Count**
0	T	528 non-null
1	N	528 non-null
2	p	528 non-null
3	k	528 non-null
4	Bin	528 non-null
5	Cost	528 non-null
6	SDratio	528 non-null
7	correl	528 non-null
8	cancor1	528 non-null
9	cancor2	528 non-null
10	fract1	528 non-null
11	fract2	528 non-null
12	skewness	528 non-null
13	kurtosis	528 non-null
14	Hc	528 non-null
15	Hx	528 non-null
16	MCx	528 non-null
17	ENAtr	528 non-null
18	NSRatio	528 non-null
19	target	528 non-null

Table 4: Summary of Dataset D2 Attributes

An investigation has been made to understand the nature of the target variable in D2, main results are:

- 55 duplicate target values
- 381 instances with a unique target value
- 147 instances that have the same target value as another

In Figures 2, 3, and 4, we visually see the results of the interpolation process on the variables with missing values. Each graph has the original and interpolated data overlapping to better appreciate the effects of this technique.

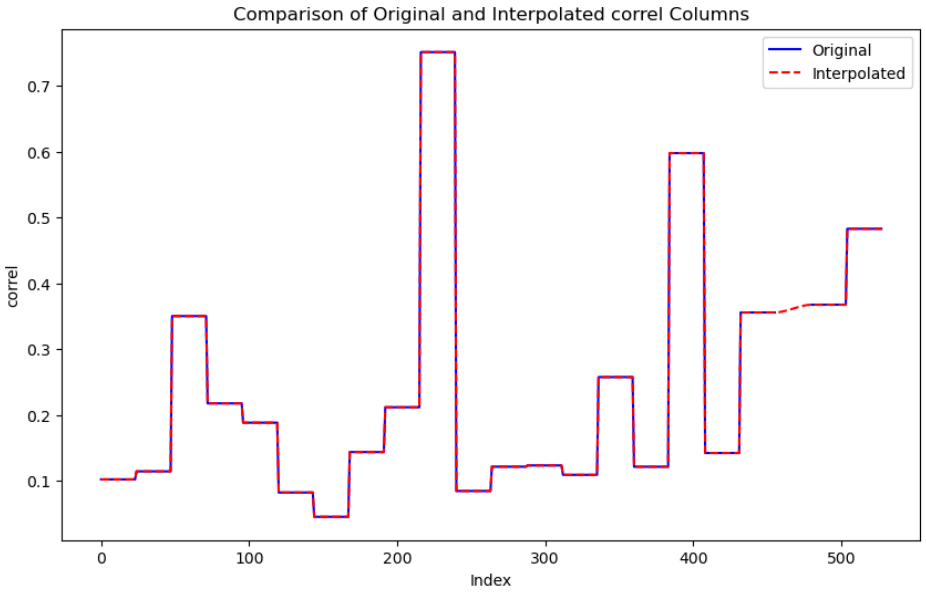

Figure 2: correl Distribution

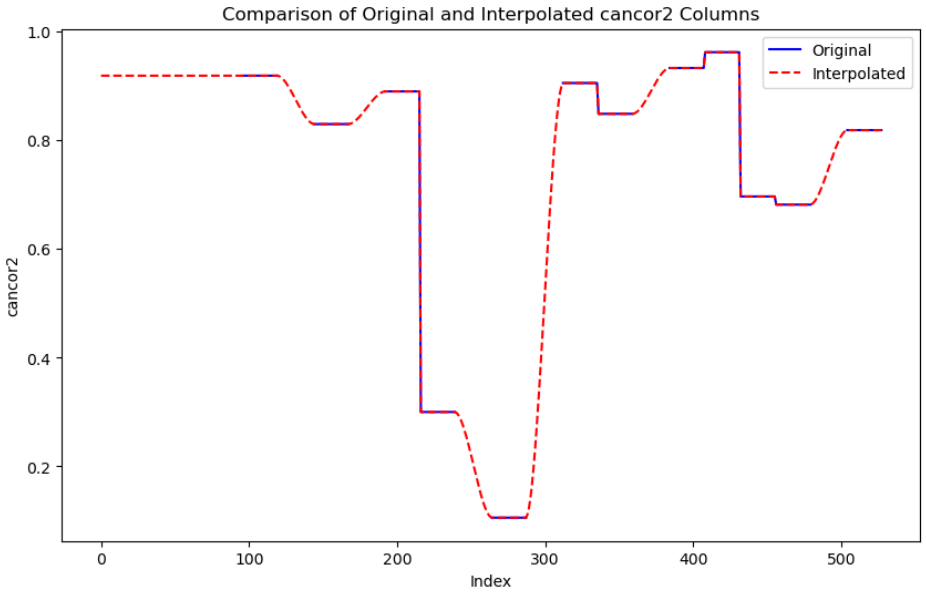

Figure 3: cancor2 Distribution

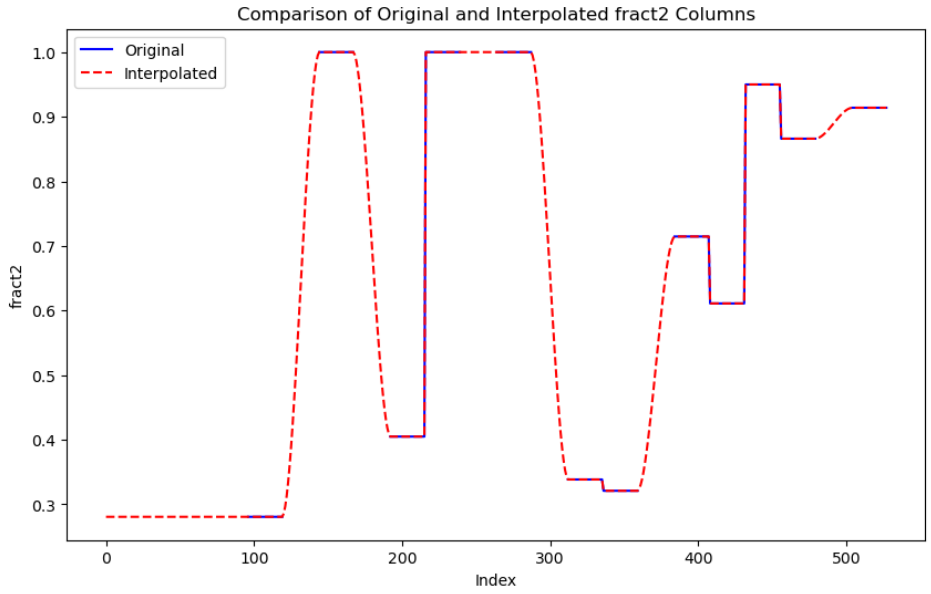

Figure 4: fract2 Distribution

Observations:
As expected the impact of the interpolation influence is more visible in cancor2 and fract2, which have 240 missing values each, over a total of 528 instances. correl only has 24 missing values, so the interpolation has a minor impact.

3.4 Standardization

To increase the models' performance, we standardize the data.
Standardization applied to a dataset, especially in the context of data processing and statistical analysis, involves rescaling the features of the data so that they have a mean of 0 and a standard deviation of 1. The main reasons for standardizing data are:

- **Feature Scaling:** StandardScaler transforms each feature to have a mean of zero and a standard deviation of one. This scaling ensures that all features contribute equally to the model's performance, preventing features with larger scales from dominating the model's learning process.
- **Uniformity in Scale:** Many machine learning algorithms, especially those involving distance calculations like SVMs, perform better when the input data is standardized. This is because standardization eliminates the bias caused by the different scales of data features.
- **Algorithm Convergence:** Standardization can speed up the convergence of gradient descent algorithms used in many regression models. By scaling the features, the optimization algorithm can navigate the solution space more efficiently, leading to faster convergence.
- **Interpretability:** Standardized data makes it easier to understand the importance of each feature since they are all on the same scale. This is particularly useful in models where feature weights indicate the importance of features.

Table 5 shows the effects of standardization in the data.

Variable	Mean Before	Mean After	Std Dev Before	Std Dev After
T	4569.04545	-2.287732e-17	5578.155240	1.0
N	10734.18182	-2.287732e-17	14247.300659	1.0
p	29.54545	-2.287732e-17	36.040113	1.0
k	9.727273	-2.287732e-17	18.929647	1.0
Bin	3.181818	-2.287732e-17	9.087237	1.0
Cost	0.136364	-2.287732e-17	0.343500	1.0
SDratio	1.479105	-2.287732e-17	0.643741	1.0
correl	0.236838	-2.287732e-17	0.181793	1.0
cancor1	0.794836	-2.287732e-17	0.152942	1.0
cancor2	0.741058	-2.287732e-17	0.257999	1.0
fract1	0.700673	-2.287732e-17	0.337779	1.0
fract2	0.700042	-2.287732e-17	0.282021	1.0
skewness	1.784218	-2.287732e-17	1.750721	1.0
kurtosis	22.667173	-2.287732e-17	40.926150	1.0
Hc	1.871577	-2.287732e-17	1.414734	1.0
Hx	3.345018	-2.287732e-17	1.764031	1.0
MCx	0.316809	-2.287732e-17	0.328075	1.0
EnAtr	20.664158	-2.287732e-17	34.874528	1.0
NSRatio	28.873002	-2.287732e-17	37.088202	1.0
target	99.552509	-2.287732e-17	764.616147	1.0

Table 5: Comparison of statistical metrics before and after standardization

We report the new min and max of the variable T to show how the values are rescaled. All variables have the same min and max.

	Min Before	Min After	Max Before	Max After
T	270.00000	-2.58395	20000.00000	15.632382

Table 6: Comparison of Min and Max Values Before and After for T

Figure 5 shows a comparative visualization of original and standardized dataset variables.

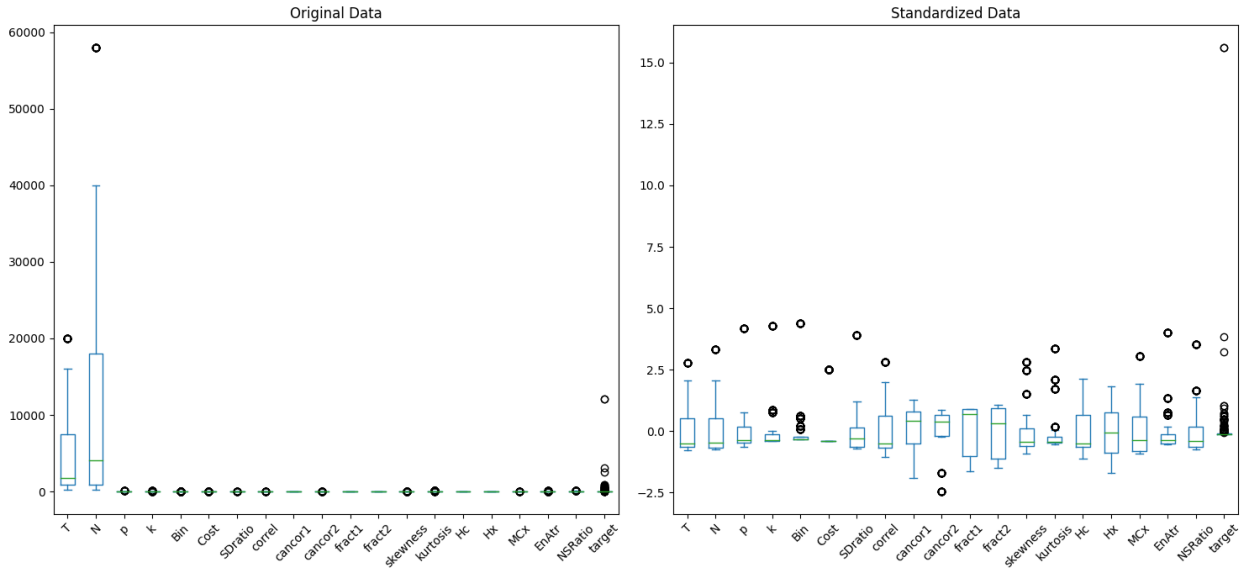

Figure 5: Comparison between original and standardized data.

Observations:

- **Scale Difference:** The original data has variables on different scales, which is evident by the wide range of values on the y-axis. After standardization, all variables are on the same scale, with a mean of 0 and standard deviation of 1. This is useful for algorithms that are sensitive to the scale of data.
- **Outliers:** The presence of outliers is more evident in the standardized data. While they were also present in the original data, standardization makes them stand out more relative to the central tendency and spread of the data.
- **Variable Comparison:** Standardization allows for the direct comparison of variability across variables. In the original data, some variables with larger scales may appear to have more variability, but after standardization, it's clear that some variables with smaller scales actually have a comparable level of variability.
- **Distribution Shape:** The shape of the data distribution for each variable is preserved after standardization. If a variable has a skewed distribution in the original data, it will remain skewed after standardization.
- **Utility for Machine Learning:** Standardized data is ready for use in algorithms such as Support Vector Machines, which assume that all features are centered around zero and have variance in the same order.
- **Data Transformation:** The standardized data plot does not have units on the y-axis because standardization removes the original units, transforming the data into dimensionless z-scores.

4 Regression

4.1 Preliminary Steps

The dataset has been split using the canonical settings, using 20% of the data as a test set. For reproducibility, we have set a seed.

4.2 Models

The project requests us to use six regression models to analyze the Meta dataset: Linear Regression, Decision Tree Regresson, Random Forest Regressor, Support Vector Regression, Gradient Booosting Regressor, and Logistic Regression. Following, an overview of them:

- **Linear Regression:**
  Linear regression is a linear approach to modeling the relationship between a dependent variable, y, and one or more independent variables denoted X. The case of one independent variable is called simple linear regression; for more than one, the process is called multiple linear regression. The linear regression model assumes a linear relationship between the input variables (X) and the single output variable (y). More specifically, that output (y) can be calculated from a linear combination of the input variables (X). When there is a single input variable (X), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.
  Linear regression is sensitive to outliers, which can significantly affect the slope and y-intercept of the regression line. It's also not suitable for non-linear relationships, where the assumption of linearity is violated.

- **Decision Tree Regressor:**
  A decision tree regressor is a non-parametric supervised learning method used for regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

  A decision tree is built by recursively partitioning the input space, and defining a local model in each resulting region of input space. This process is known as recursive binary splitting. The tree is constructed in a top-down manner, splitting the dataset into two or more homogeneous sets based on the most significant splitter/differentiator in input variables.

  Advantages:
  1. Easy to interpret and visualize.
  2. Can handle both numerical and categorical data.
  3. Requires little data preprocessing—no need for normalization.
  4. The non-parametric nature means that the model is not constrained by a particular form of function.

  Disadvantages:
  1. Sensitive to noisy data, which can lead to overfitting.
  2. Small changes in the data can result in a completely different tree.
  3. Decision tree learners can create overly complex trees that do not generalize well from the training data (overfitting).

- **Random Forest Regressor:**
  A Random Forest regressor is an ensemble learning method for regression that operates by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees. Random Forest combines the simplicity of decision trees with flexibility, resulting in a vast improvement in accuracy.

  The Random Forest regressor builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Advantages of Random Forest:
1. It can handle large datasets with higher dimensionality.
2. It provides higher accuracy through cross-validation.
3. Random Forest can handle missing values and maintain accuracy for missing data.

Disadvantages include:
1. Complexity, as it is more computationally intensive than decision trees.
2. Less interpretability, as it is harder to visualize the model due to the presence of numerous trees.

Random Forests also tend to perform well against overfitting as the model uses the average of all trees, which cancels out biases. The model is robust to outliers and can handle non-linear data efficiently.

- **Support Vector Regression (SVR):**
SVR is used for its effectiveness in high-dimensional spaces and its flexibility in kernel choice, which allows for modeling complex, non-linear relationships.
Support Vector Regression (SVR) applies the principles of Support Vector Machines (SVM) to regression problems. It attempts to find a function $f(X)$ that deviates from y_i (actual values) by a value no greater than ϵ for each training point X_i, and at the same time is as flat as possible.

SVR has several advantages:
- Effective in high-dimensional spaces.
- Still effective in cases where the number of dimensions exceeds the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

The disadvantages of SVR include:
- If the number of features is much greater than the number of samples, the method is likely to give poor performances.
- SVR does not directly provide probability estimates, which are desirable in many applications.
- The choice of the kernel and regularization parameter can have a large impact on the performance of the algorithm.

- **Gradient Boosting Regressor:** This model is included for its prowess in handling various types of data and its ability to optimize different loss functions, making it a powerful tool for regression tasks. It constructs a predictive model in the form of an ensemble of weak predictive models.

Advantages of GBR:
- Often provides predictive accuracy that cannot be beaten.
- Lots of flexibility - can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible.
- No data pre-processing required - often works great with categorical and numerical values as is.
- Handles missing data.

Disadvantages of GBR:
- GBR models will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting.
- Computationally expensive - GBR often requires many trees which can be time and memory exhaustive.
- The high number of hyperparameters requires careful tuning for the model to perform optimally.
- Less interpretative in nature, although tools such as feature importance scores can help to interpret the model.

- **Logistic Regression**: Despite being included in the list of methods, this method is not suitable for our regression task. The inability to use logistic regression effectively with the Meta dataset can be attributed to several key factors related to the nature of both the logistic regression model and the dataset itself. The main reasons are highlighted here

  – **Nature of The Target Variables:**
  - Logistic regression is fundamentally designed for binary or categorical outcomes. It models the probability that a given input point belongs to a certain category, typically a binary outcome (like 0/1, yes/no).
  - The dataset Meta has a continuous target variable. Logistic regression is not suitable for modeling continuous variables since it cannot predict a range of numeric values, only the probability of belonging to a class.

  – **Output Interpretation:**
  - Logistic regression outputs probabilities, not continuous numbers. It predicts the likelihood of an observation belonging to one of the classes in a classification problem.
  - The target variable is continuous and not categorical, using logistic regression would be inappropriate as it would not provide meaningful predictions. It cannot capture the nuances of the range of continuous data.

4.3 Error Metrics

The models' performance should be measured through three error metrics, MAE, MAPE, and SMAPE.

- **MAE (Mean Absolute Error):**
  Represents the average absolute difference between the predicted and actual values. Lower values indicate better performance.
  Interpretation:
  - Lower MAE: A lower MAE value indicates a more accurate model, as it means the predictions are closer to the actual values.
  - Units: MAE is expressed in the same units as the data, making it easy to interpret.
  - Robustness: MAE is not sensitive to outliers. Large errors don't disproportionately affect the metric.
  - Zero MAE: If MAE is zero, the model is perfect, which is rare in practice.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

- **MAPE (Mean Absolute Percentage Error):**
  Expresses accuracy as a percentage of the error, which can be particularly helpful for understanding the magnitude of the error in relation to the actual values. Again, lower values are better. Interpretation:
  - Lower MAPE: A lower MAPE value indicates a more accurate model. The closer MAPE is to 0%, the better the model's predictive accuracy.
  - Scale-Independence: MAPE is expressed as a percentage, which makes it easy to interpret regardless of the scale of the data.
  - Limitations: MAPE can be misleading if there are actual values very close to zero, and it cannot be used if there are actual values that are exactly zero. It also does not distinguish between over-predictions and under-predictions.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{2}$$

- **SMAPE (Symmetric Mean Absolute Percentage Error):**
  Similar to MAPE but symmetric, taking the absolute difference between the predicted and actual divided by the sum of the absolute values of the predicted and actual. Lower values indicate better predictions. Interpretation:

- Lower SMAPE: A lower SMAPE value indicates a more accurate model. Values closer to 0% are ideal.
- Scale and Symmetry: SMAPE is expressed in percentages and is more symmetric compared to MAPE, meaning it treats overestimation and underestimation more equally.
- Handling Zeros: SMAPE can better handle cases where actual values are zero or close to zero, though it's not perfect in this regard.

$$\text{SMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)} \tag{3}$$

While MAE and MAPE are already implemented in the scikit-learn library, for SMAPE a custom function has been created based on the formula (3) which we analyzed during the study.

4.4 Results

To ensure reproducibility, a seed has been set before training the models.
The models have been trained with the default settings, and no fine-tuning has been made.

Table 7 displays the results of the models being trained on the two datasets.

Table 7: Regression Results on Datasets D1 and D2

Model	**MAE**	**MAPE**	**SMAPE**
Results on D1			
Linear Regression	0.30068	1.13869	0.31693
Support Vector	0.29498	0.64266	0.46977
Decision Trees	0.30068	1.13869	0.31693
Random Forest	0.30321	1.17923	0.32903
Gradient Boosting	0.30068	1.13867	0.31692
Results on D2			
Linear Regression	0.37144	1.13210	0.37153
Support Vector	0.40552	0.93828	0.64744
Decision Trees	0.36547	1.11986	0.31855
Random Forest	0.37528	1.16782	0.33582
Gradient Boosting	0.36338	1.06534	0.31938

Results are discussed in section 5.

4.5 Graphical Representations

To better understand the results we have plotted them with 3 types of plots, each of which highlight from a different perspective the results.
Figure 6 shows a bar chart highlighting the performances of each model, in terms of MAE, MAPE, and SMAPE, on both datasets. This type of chart is useful for comparing multiple regression models, across the different metrics, highlighting the differences model's performance for each dataset.

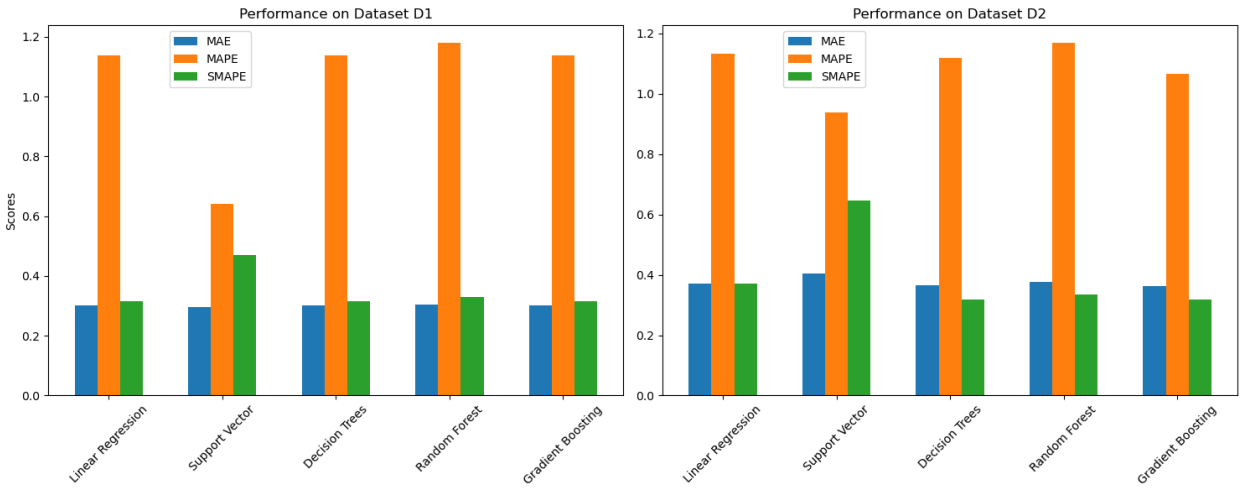

Figure 6: Performance on Dataset D1 and D2.

Figure 7, like the previous one, is a bar chart, but it emphasizes the error metrics for each model, on both datasets, from a different perspective.

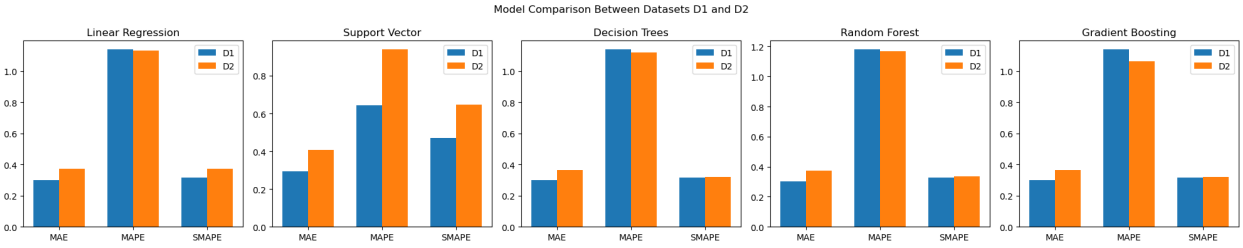

Figure 7: Model Comparison Between Dataset D1 and D2.

Last but not least, in Figure 8, we used line charts to compare the performance of the regression models across three metrics. Each line represents one of the metrics, and points on the lines correspond to the performance scores of the models.

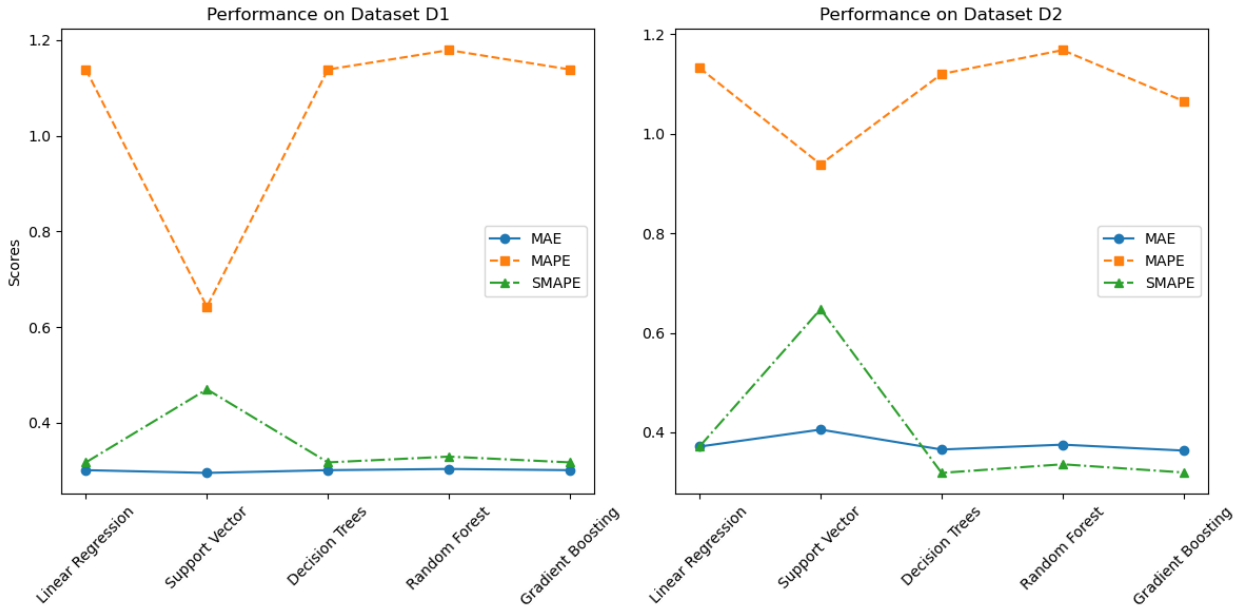

Figure 8: Performance on Dataset D1 and D2.

4.6 Feature Importance

Additionally, we have explored the features to understand how they influence the results.
Feature importance refers to the techniques for assigning scores to input features based on how useful they are at predicting a target variable. Understanding feature importance can lead to insights into the dataset and the model's behavior. For example, Knowing which features are less important can lead to model simplification by removing these features. This can reduce model complexity, improve generalization, and decrease overfitting. By the fact we have different datasets, we can analyze how the features can influence the same model based on the size and type of the data.
Two techniques have been used, based on the model:

- **Coefficient in Linear Models**: In linear models (like Linear Regression), the coefficients can be used to represent the importance of features. The size and sign of the coefficients indicate the extent and direction of the impact on the target variable.
- **Tree-based Models**: Models like Decision Trees, Random Forests, and Gradient Boosting Machines have a built-in 'feature_importances_' attribute.

Figure 9 shows the importance of each feature, for each model, in both datasets:

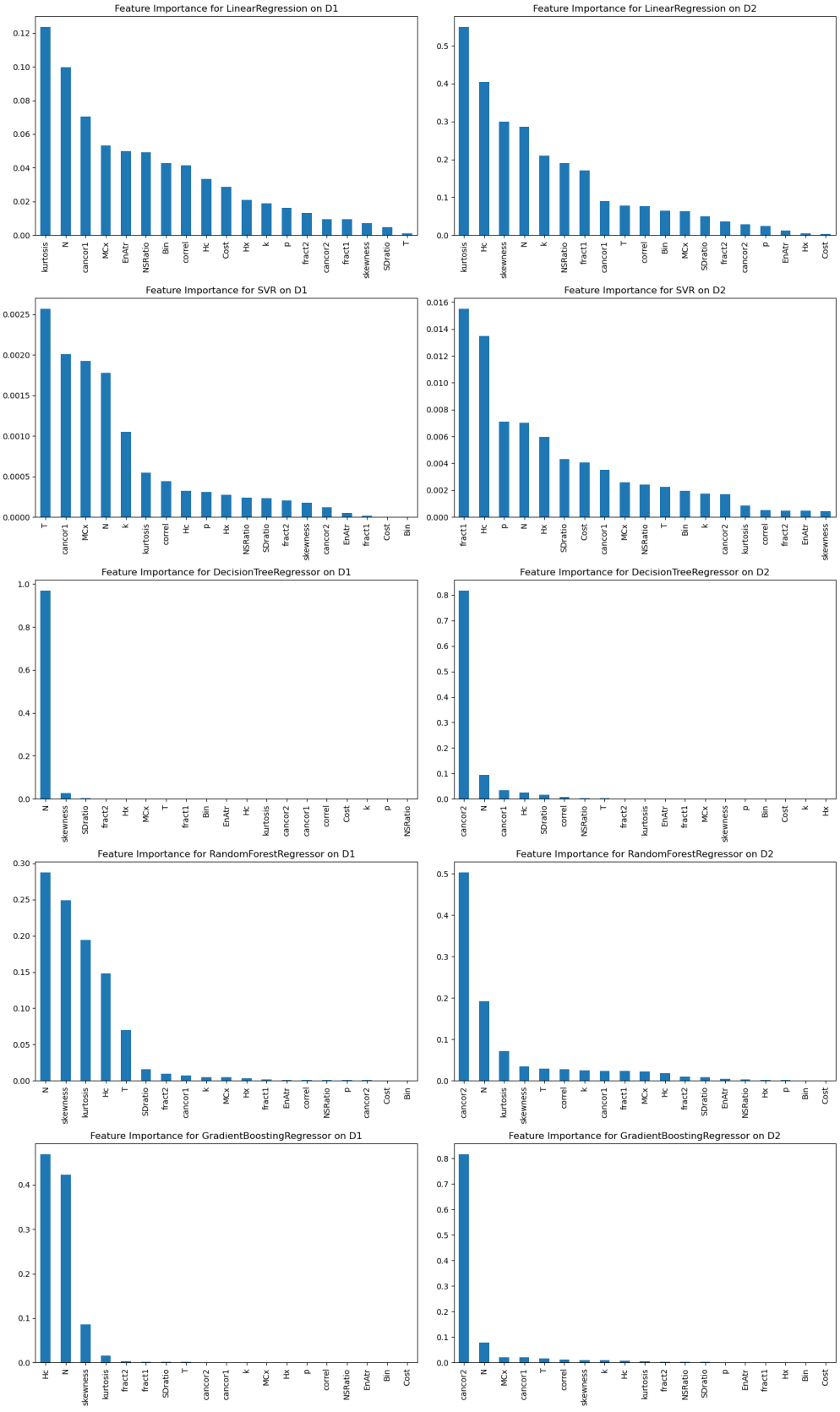

Figure 9: Feature Importance

Observations:
For Linear Regression, kurtosis is the main feature on both datasets along with N. The latter, however, is a bit important for interpolated data. With SVR, the importance of attributes changes dramatically from one dataset to another. In decision trees, N dominates when we don't introduce noise, while in interpolated, cancor2 data it is dominant. To consider in this case cancor2 is one of the variables that has been interpolated. The same scenario is also found with Random Forest and Gradient Boosting, which have the same structure as the Decision Tree. Linear Regression is the one that uses the most attributes to make decisions, followed by SVR. The other models base their decisions on a few variables. Overall N is the attribute that on average influences the decisions of models the most. Variables with the greatest influence predominantly have values near zero, which could adversely affect performance.

5 Discussion and Conclusions

In this final section, we analyze the regression results, we can discuss the performance of the models on datasets D1 and D2. We also provide a summary with final considerations on tasks and results.

5.1 Dataset D1 Analysis

On dataset D1, the models displayed relatively uniform MAE values, suggesting that the average error magnitude of the predictions was consistent across models. However, there were differences observed in MAPE and SMAPE values, which indicate variability in error proportionality relative to the true values. **Linear Regression** has moderate error metrics, which is expected given the smaller dataset size, potentially leading to a more tailored fit. **Support Vector** model has the lowest MAPE, suggesting it is better at dealing with percentage errors relative to true values in a smaller dataset. **Decision Trees** and **Gradient Boosting** share the exact MAE and very close MAPE values, but both have the lowest SMAPE, indicating effective handling of symmetrical errors. The **Random Forest** model performs competitively with Linear Regression and Gradient Boosting in terms of MAE and SMAPE but has a higher average percentage error as shown by MAPE. This could imply that Random Forest may not be the best model for this dataset if the percentage error is a critical measure, but it's relatively strong in other error metrics.

5.2 Dataset D2 Analysis

The results on dataset D2 show a general increase in error metrics across all models when compared to D1, which could be attributed to the complexity or noise within the dataset. Despite the increase, the Gradient Boosting model showed the lowest MAE and MAPE, pointing to its robustness and ability to handle noise or complex data structures. **Linear Regression** shows an increase in MAE and SMAPE compared to D1, possibly struggling with the larger dataset size and interpolated values. **Support Vector** has a notably higher SMAPE on D2, indicating challenges with the interpolation and a potential increase in both overestimations and underestimations. **Decision Trees** show an improvement in MAPE, suggesting that the model is less sensitive to the increased dataset size and interpolated values. **Random Forest** and **Gradient Boosting** both have increased MAE while maintaining a relatively stable SMAPE, demonstrating their robustness to interpolated data. The highest MAE values could be attributed to the high number of predictions closed to zero.

5.3 Conclusions

The final consideration of this analysis would focus on the consistency of model performance across different metrics and datasets. The similarity in MAE values between Linear Regression and Gradient Boosting could imply that for this particular task, the complexity added by Gradient Boosting may not necessarily translate to better performance, at least in terms of MAE. However, the varied performance in MAPE and SMAPE suggests that a deeper dive into the type of errors and their relativity to the actual values is necessary. It's also evident that model performance can vary significantly across datasets, which underlines the importance of understanding dataset characteristics and possibly tailoring models to specific data scenarios. The insights gained from this analysis could guide further model tuning, feature engineering, or even the collection of additional data to improve model robustness. In summary, the evaluation of model performance suggests that while some models may excel in minimizing the average error, they may not proportionally minimize errors relative to the scale of actual values. The Support Vector model, despite its low MAE on D1, may not be the best choice when the relative size of the error is considered, as indicated by its higher SMAPE. Conversely, Gradient Boosting showed a strong performance on D2, suggesting that it could be a preferred model in more complex scenarios. This analysis underscores the importance of considering multiple metrics to fully understand model performance and the necessity to tailor model selection to the specific characteristics of the dataset at hand.

5.4 Summary

Comprehensive Analysis

- **Data Understanding and Preprocessing**
  - The dataset contains a 'target' variable of interest, with 'correl', 'cancor1', and 'fract2' having missing values.
  - 'DS_Name' and 'Alg_Name' are categorical variables, and they were excluded from the analysis.
- **Datasets Overview**
  - D1: Missing values were handled by deletion. This may have reduced the dataset's size and potentially impacted representativeness and model performance.
  - D2: Missing values were interpolated, aiming to preserve information but potentially introducing noise or bias into the dataset.
- **Model Evaluation**
  - Models were assessed using MAE, MAPE, and SMAPE, with visual comparisons illustrating their performance on D1 and D2.
- **Metrics Interpretation**
  - MAE shows average absolute errors; lower values are better.
  - MAPE gives errors as a percentage, where lower is better.
  - MAPE accounts for symmetry in errors, with lower values indicating better predictions.
- **Modeling Insights**
  - D1: Clean dataset with missing values handled by deletion.
  - Support Vector Machine performs best in MAE and MAPE, suggesting accuracy with smaller datasets.
  - D2: Larger dataset with interpolated missing values.
  - Gradient Boosting shows the best MAE, indicating effectiveness in managing absolute errors even with interpolated data.
  - Decision Trees perform well in terms of SMAPE, showcasing their strength in managing relative errors effectively.
- **Overall Assessment**
  - Gradient Boosting and Decision Trees demonstrate resilience and robustness across both datasets.
  - Data preprocessing, such as handling missing values through interpolation or deletion, significantly influences model performance.
  - Feature engineering, hyperparameter tuning, and ensemble methods could further enhance model effectiveness.
- **Recommendations for Practice**
  - Select Support Vector Machine for smaller, cleaner datasets (D1) for its lower MAE and MAPE.
  - Consider Gradient Boosting or Decision Trees for larger datasets with interpolated values (D2), focusing on the balance between MAE, MAPE, and SMAPE.

6 Additional Tests and Future Improvements

In this section, we give suggestions for further improvements, as well as a high-level description of the not mandatory test we have conducted.

6.1 Additional Tests

Additional investigation has been made to reinforce the choice made during the project. Following is the list of the tests, which can be optionally shown in the jupyter notebook:

- Evaluation with MSE (Mean Squared Error)
- Evaluation with RMSE (Root Mean Squared Error)
- Normalization of the dataset
- Additional distribution plots

6.2 Future Directions

- Search for outliers and treat them properly.
- Investigate the impact of different interpolation methods on D2's data quality.
- Explore additional validation methods for a thorough assessment of model stability and generalizability.
- Employ cross-validation and other validation techniques for a comprehensive model evaluation.