Data Mining Project
Meta Dataset Regression

Marco Sau
60/79/00028

February 2024

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 objectives of this project are:

- Apply regression techniques.

Related objectives are:

- To perform an in-depth analysis of the Meta dataset.
- To draw meaningful insights and conclusions from the analysis.

2 Dataset

2.1 Description

Source: 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 mainly created for classification tasks, not for regression tasks. It's crucial to remember this key when analyzing the data and models, as it can cause issues during the process.

2.2 Main Features

The dataset has 22 features and 528 rows, of which 20 are numerical and 2 categorical. The list of charges 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 include:

- Presence of a target variable, which is the focus of the regression models.
- Three variables with missing values:
  - 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.
- DS_Name and Alg_name are the only categorical attributes.

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

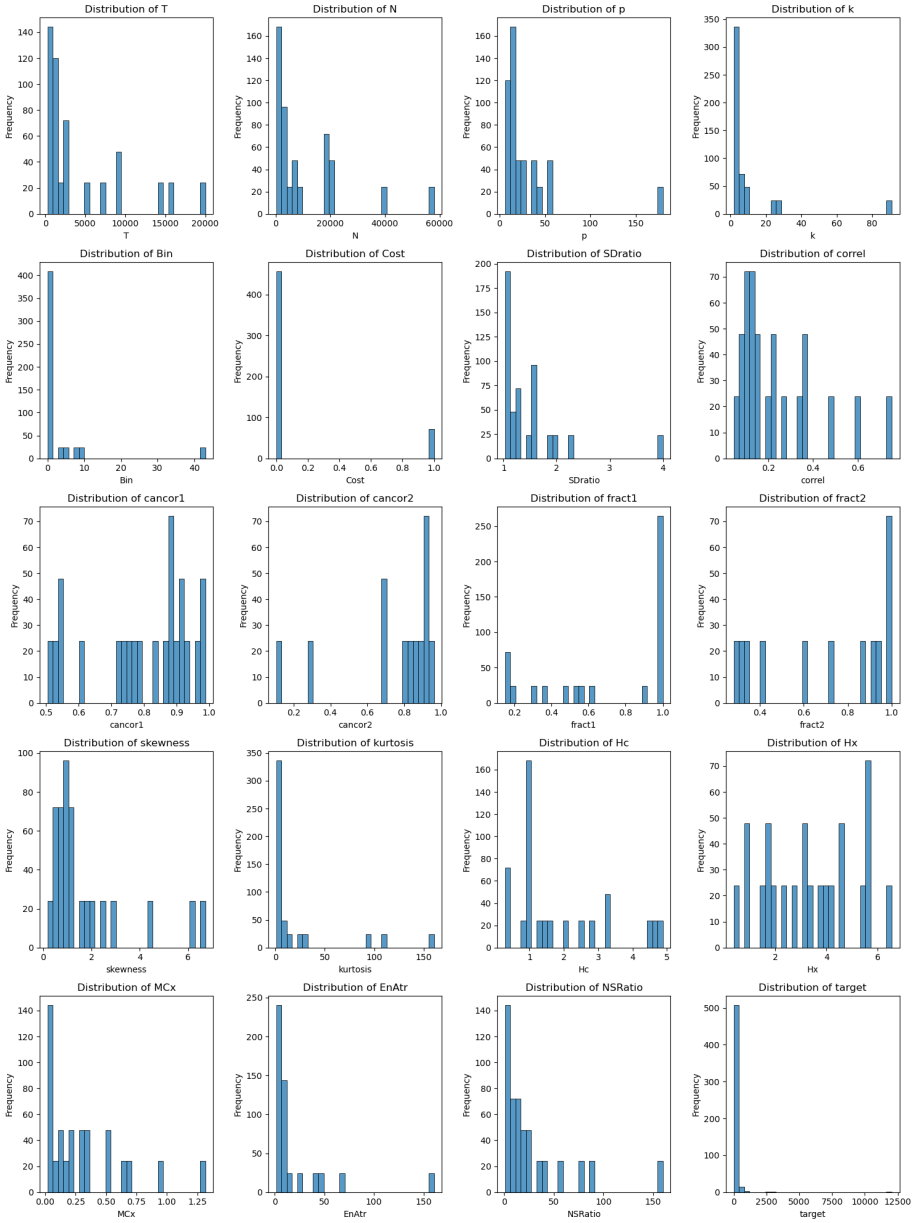

Figure 1: Numeric Distribution

3 Preprocessing

3.1 Removing Unnecessary Features

The first step was organizing the data in a pandas DataFrame. During this process, we removed the nominal features ('DS_Name' and 'Alg_Name') as not relevant for regression models.

3.2 Handling Missing Values

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

3.2.1 D1: removing rows with missing values

We have deleted all rows with missing values. This approach, while simple, can lead to the loss of important information, especially if the missing data is numerous.
The new dataset now has 264 rows, which is exactly half of the size of the original dataset.

3.2.2 D2: interpolate missing values

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 you apply this method 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.

In Figures 2, 3, and 4, we can 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.

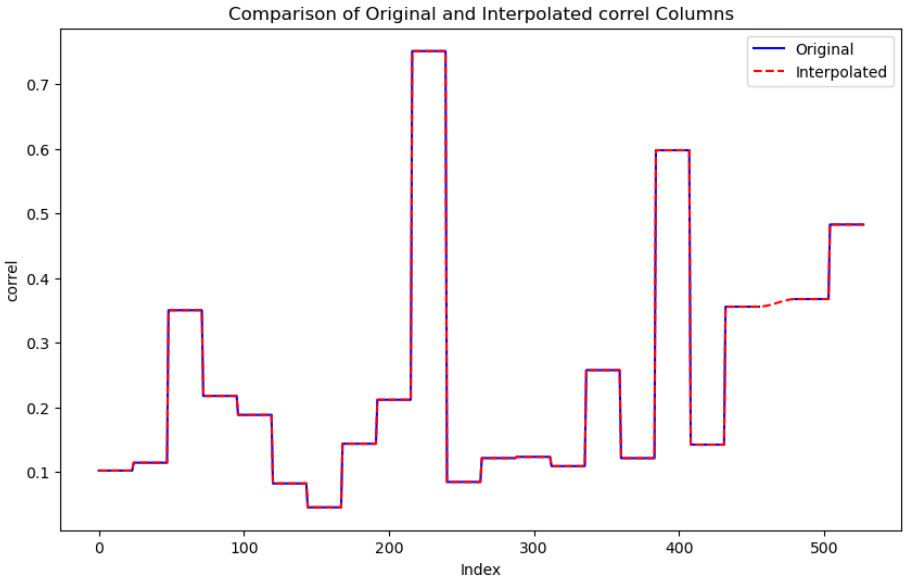

Figure 2: correl Distribution

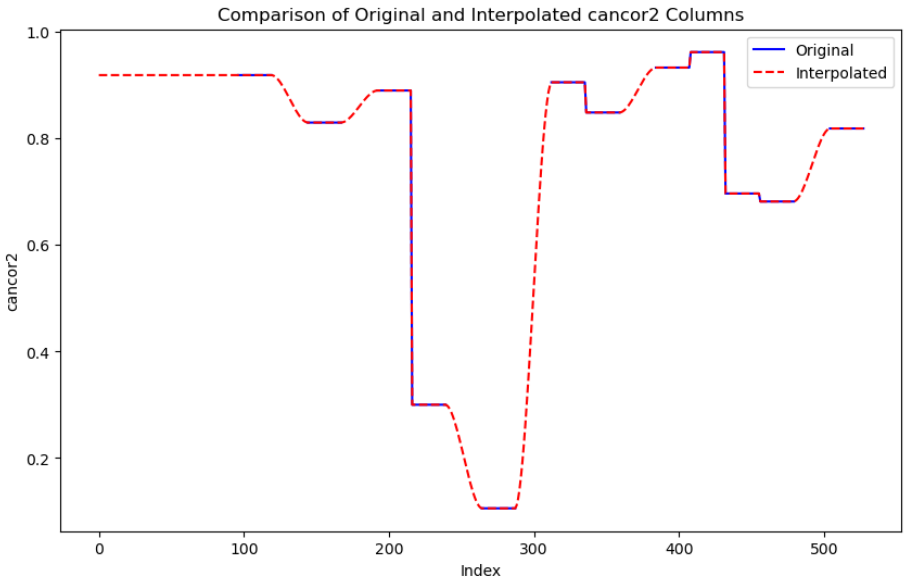

Figure 3: cancor2 Distribution

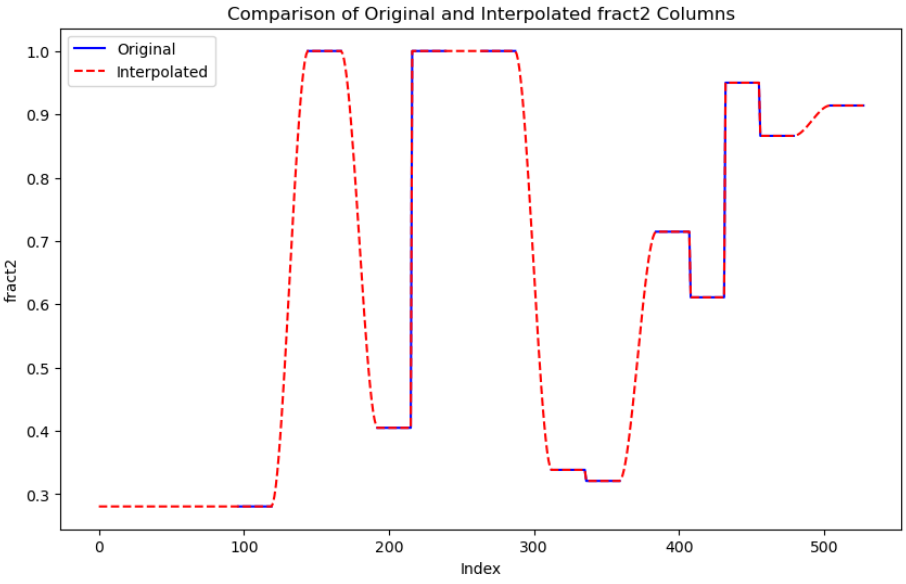

Figure 4: fract2 Distribution

3.3 Standardization

To increase the performance of the models we perform a standardization of the data.
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.

4 Regression

4.1 Models

This project employs a variety of regression models to analyze the Meta dataset. Each model is chosen for its unique characteristics and suitability for different types of regression tasks. The models used include:

- **Linear Regression:** As a fundamental model in statistical modeling, linear regression is used for its simplicity and efficiency in establishing a linear relationship between the target variable and predictors.
- **Decision Tree Regressor:** This model is used for its ability to capture non-linear relationships. It's beneficial for understanding the feature interactions and provides a clear interpretation of how decisions are made.
- **Random Forest Regressor:** An ensemble model comprising multiple decision trees, Random Forest is chosen for its robustness and ability to reduce overfitting.
- **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.

- **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.

- **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.

4.2 Error Metrics

- **MAE:** Represents the average absolute difference between the predicted and actual values. Lower values indicate better performance.

- **MAPE:** Expresses accuracy as a percentage of the error. Again, lower values are better.

- **SMAPE:** 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.

While MAE and MAPE are already implemented in the scikit-learn library, for SMAPE a custom function has been created based on the formula analyzed during the study. The details of such a function are well described in the notebook.

4.3 Justification for Model Choices

The selection of these models is based on their diverse capabilities in capturing different aspects of the data. Linear Regression provides a baseline, while Decision Tree and Random Forest offer insights into feature interactions. SVR and Gradient Boosting are capable of modeling complex patterns, making them suitable for sophisticated regression tasks.

4.4 Results and Graphical Representations

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 applied.

4.4.1 Results

Table 2 displays the results of the models being trained on the two datasets.

Table 2: 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

Observations on D1:

- 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.

Observations on D2:

- Linear Regression shows an increase in all error metrics 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 and MAPE but maintain a relatively stable SMAPE, demonstrating their robustness to interpolated data.

4.4.2 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 5 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.

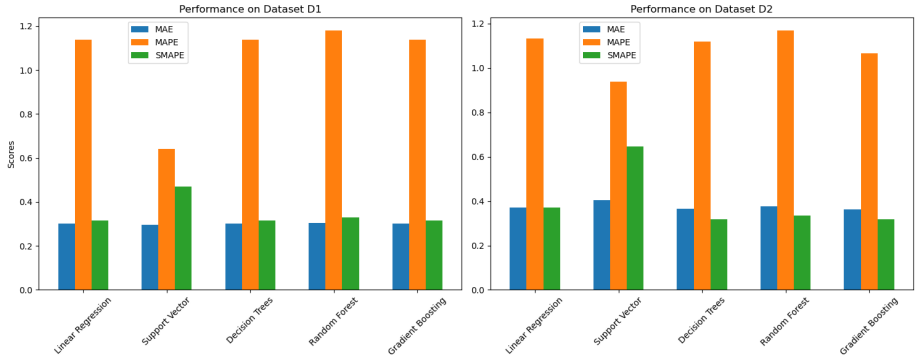

Figure 5: Performance on Dataset D1 and D2.

Figure 6, like the previous one, is a bar chart, but it emphasizes the error metrics for each model, on both datasets, from a different perspective.

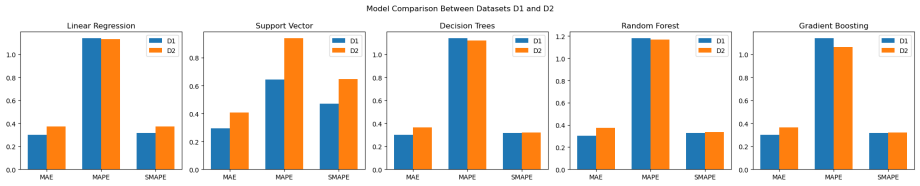

Figure 6: Model Comparison Between Dataset D1 and D2.

Last but not least, 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 6, shows the results:

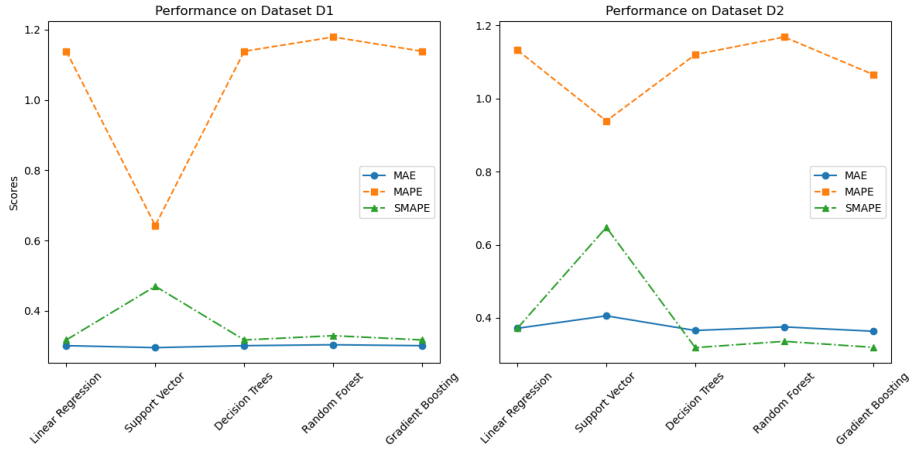

Figure 7: Performance on Dataset D1 and D2.

4.4.3 Feature Importance

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 Regres-

sion), 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.

A deep analysis has been made to understand how each attribute influences the performance. Figure 5 shows the importance of each feature, for each model, in both datasets:

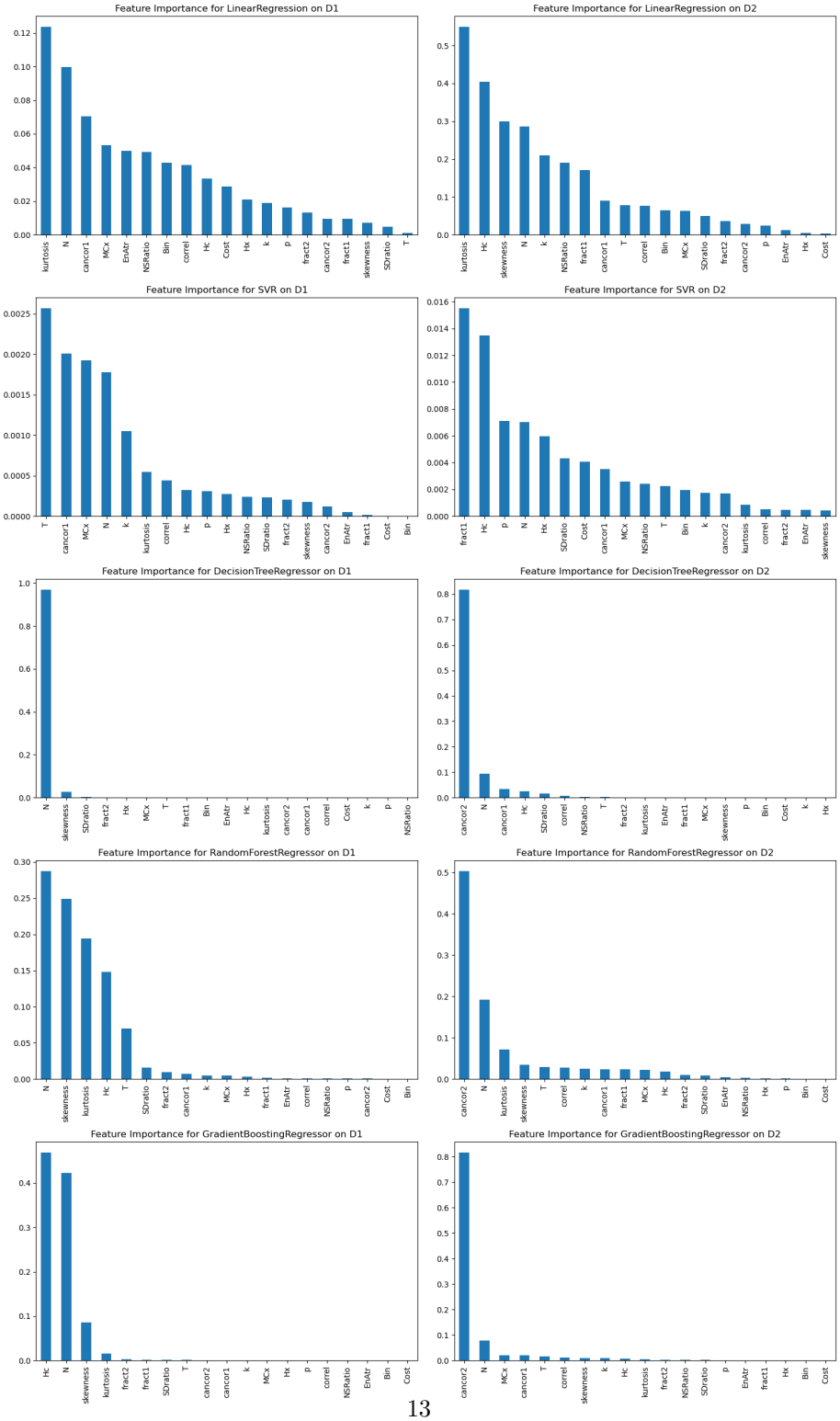

Figure 8: Feature Importance

Feature Importance Analysis: As for Linear Regression, the kurtois 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.

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. The models are Linear Regression, Support Vector Regression (SVR), Decision Trees, Random Forest, and Gradient Boosting, and the error metrics used are MAE, MAPE, and SMAPE.

6 Models Performance Analysis

The performance of regression models on the datasets D1 and D2 was evaluated using various metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). These metrics provide insights into the models' accuracy and reliability in predicting the target variable.

6.1 Dataset D1 Performance

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. The Support Vector Regression model exhibited the lowest MAE, indicating a strong predictive accuracy on average, but its higher MAPE value suggests larger errors relative to the actual value scales.

6.2 Dataset D2 Performance

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.

6.3 Comparison Between Datasets

When comparing the models' performance between the two datasets, it is clear that dataset D2 was more challenging for the models, with higher error rates recorded. This could imply that D2 may have characteristics such as higher variability or less signal in the data, which complicates the modeling process.

6.4 Summary

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 MAPE. 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.

6.5 Performance Metrics

- MAE (Mean Absolute Error) is a measure of the average magnitude of errors in a set of predictions, without considering their direction. It's a straightforward metric that provides a quick look at the overall error magnitude
- MAPE (Mean Absolute Percentage Error) expresses accuracy as a percentage of the error and is particularly useful when you need to compare the performance of models across different scales.
- SMAPE (Symmetric Mean Absolute Percentage Error) addresses some of the problems with MAPE when dealing with zero or close to zero actual values, as it is symmetrical for both the forecast and the actuals.

6.6 Discussion

From the results, we observe that across both datasets, the performance of the models in terms of MAE is fairly consistent, with Linear Regression and Gradient Boosting having identical MAE values. This could indicate that these models have reached a similar level of optimization for the given datasets.

The MAPE and SMAPE values offer more diversity in performance, which suggests that the scale of the errors relative to the actual values varies more significantly between models. For instance, SVR shows a higher MAPE on D1, indicating that while its errors might be low in magnitude, they are significant relative to the scale of the actual values.

6.7 Dataset Comparison

When comparing the results from D1 to D2, we can infer that the models generally performed worse on D2. This could be due to a variety of factors, such as a difference in the distribution or complexity of the data, or perhaps D2 has more noise or outliers that impact model accuracy.

6.8 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.