Acta Univ. Sapientiae, Informatica, **V, I** (202z) pp–qq

DOI: 10.2478/ausi-20yy-nnnn

Towards better decision support in demand forecasting: global feature importance for multi-series tree-based models

Mátyás Kuti-Kreszács
Babeş–Bolyai University, Faculty of Mathematics and Computer Science, M. Kogălniceanu 1, 400084, Cluj-Napoca, Romania
e-mail: matyas.kuti@ubbcluj.ro
orcID: 0009-0004-4997-2000

Laura Dioşan
Babeş–Bolyai University, Faculty of Mathematics and Computer Science, M. Kogălniceanu 1, 400084, Cluj-Napoca, Romania
e-mail: laura.diosan@ubbcluj.ro
orcID: 0000-0002-6339-1622

Zalán Bodó
Babeş–Bolyai University, Faculty of Mathematics and Computer Science, M. Kogălniceanu 1, 400084, Cluj-Napoca, Romania
e-mail: zalan.bodo@ubbcluj.ro
orcID: 0000-0002-4857-878X

Abstract.

Global feature importance methods are one of the core tools for interpreting the role of explanatory variables in machine learning models, however, using them more complex forecasting tasks involving multiple time series can be challenging. This study focuses on tree-based ensemble models, applied to multi-series product demand forecasting. To evaluate the different global feature importance methods, we generate simulated datasets with controlled dependencies on lagged values and external demand drivers. We compare model-specific and model-agnostic global importance methods, including SHAP values, permutation importance, and tree-specific gain- and split-based importance. Our analysis focuses on uncovering pitfalls in applying these methods, including problems introduced by auto-correlation and feature scaling. This work provides actionable guidance for practitioners seeking to apply these methods in real-world forecasting scenarios and to leverage explainability methods for informed decision-making.

Key words and phrases: tree-based demand forecasting, multi-series, feature importance.

1 Introduction

Product demand forecasting is a common business problem in many industries, but especially in production planning, manufacturing, logistics, inventory management, retail, and marketing. Machine learning (ML) models are often used as a tool to solve these problems, but achieving the highest accuracy is not always the primary objective: in some cases, it is more important to understand the underlying factors that drive the forecast so that the business can make better decisions or even inspect how changing some of the factors would affect the forecast. Consequently, there is a growing interest in explainable AI (XAI) to tackle these issues [1]. Unfortunately, most of the time demand forecasting is not as simple as forecasting a single time series. The demand for products has to be forecasted for multiple geographical regions, different types of selling points, or multiple brands and stock-keeping units (SKUs) of the same product. This is known as multi-series forecasting through a global forecasting model, which is applicable when time series share common patterns, which can be leveraged also to improve model generalisation. The complexity of the problem increases even more since demand is typically influenced by a variety of internal and external factors, such as weather, promotions, holidays, or even economic and demographic indicators.

Complex ML models, including ensemble models and neural networks, are typically used to solve these forecasting problems. However, one of the main concerns with these models is that they are opaque (black-box) models, which means they are hard to interpret, which would be crucial for decision support and evaluating if we can trust the models' predictions. Given an already trained black-box ML model, one of the most commonly used methods for post-hoc model explainability is feature importance (FI) estimation. It helps quantify how explanatory variables, such as lagged values and external factors, influence the predictions of the model. For demand forecasting, tree ensembles are a common choice [2], as they can capture complex patterns; meanwhile, they have intrinsic feature importance estimation methods in addition to model-agnostic methods that can be applied to any model.

Our research tackles some of the challenges faced during the opaque tree-ensemble models' interpretation, focussing on the global feature importance estimation in the context of multi-series forecasting models. We describe some of the challenges we also faced in applying global feature importance methods while working on [3]. We design a set of controlled experiments with simulated time-series data to uncover the pitfalls and provide practical recommendations for leveraging them in decision support, contributing also to methodological advancements for real-world applications.

Our work is stuctured on the following way; in Section 2, we provide a literature review of feature importance methods and their some of the challenges in application for multi-series forecasting models. Then in Section 3, we describe the methodology used to generate the simulated datasets for multi-series forecasting, the feature importance methods and how to address some of the challenges. In Section 4, we present the experimental results, from data generation to feature importance estimation. Then in Section 5, we describe the practical recommendations, and limitations of the study, and finally, in Section 6, we present the conclusion of the study and future research directions.

2 Background and Related Work

XAI methods have a broad range of applicability in different domains like healthcare, finance or autonomous driving, where validation of ML models and their predictions is essential. In healthcare, they can be used as part of clinical decision support systems [4], in finance for credit risk assessment [5] or in autonomous driving for safety-critical systems [6].

XAI lately attracted more attention, as regulations like the European Union's AI Act [7] require systems using AI to be more transparent, while relevant information should be provided to explain the decision-making process. There are also applicability scenarios for XAI in demand forecasting, focused both on reliable predictions, understanding buying behaviour, and supporting simulations of "what if" scenarios [1]. Some XAI methods are targeting multiseries and multivariate time series [8], [9] however, the study of multi-series forecasting models' interpretability is still limited.

2.1 Feature importance methods

Feature importance (FI) is an interpretation technique that provides a summary statistic that assigns a relevance score for each input feature of a machine learning model [10]. FI methods can be categorised on the basis of different aspects as in Fig. 1, the most common being their scope, in which case they can be global or local [10], [11]. Global FI (GFI) methods or model feature attribution methods explain the contribution of features to overall predictions, whereas local FI measures the impact of individual features on specific predictions [10]. These methods can be model-specific, which are limited to specific model types, while model-agnostic ones are applicable independent of the model type [10]. A third way to categorise GFI methods is by the approach of calculation, in which case the importance can be based on the model's structure, while the other approach relies on a dataset.

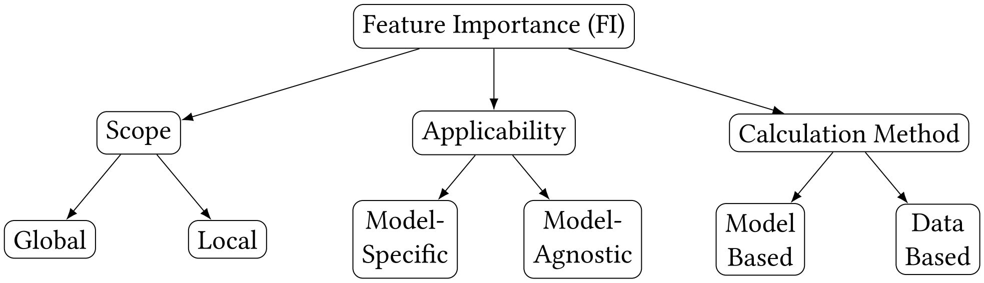


Figure 1: Categorization of feature importance methods (based on[10])

2.1.1 Model-agnostic feature importance methods

Commonly used model-agnostic methods are permutation feature importance (PFI) and SHapley Additive exPlanations (SHAP) [12], [13]. PFI method was introduced in [14] to evaluate the importance of features in random forests. It's a model-agnostic, data-based method that measures the importance of features by the decrease of model predictive performance when the feature is permuted. The goal of permutation is to break the relationship between the feature and target variable and observe the change in the model's performance – in case the feature is important, the model's performance will decrease significantly. Given that the performance of the model can be measured by different metrics, such as mean squared error (MSE), mean absolute error (MAE) or coefficient of determination (R^2), the importance of a feature can be calculated as the difference between the performance of the model in the original data and the permuted data. Several limitations exist, starting with its sensitivity to overfitting and underfitting in the model [15], so it is advisable to apply it to both the train and the test data sets. Furthermore, another flaw of the PFI method is that it can generate cases in which the model does not have training data or is not possible in real scenarios [16], [17]. It is also sensitive to correlated features, in which case the importance of the correlated features is underestimated. In this case, it is recommended to permute the correlated features together to get more accurate results [15] and also to use the conditional permutation importance method [18]. Conditional permutation importance [18] is a method that addresses the issue of correlated features in the PFI method. The difference between PFI and conditional permutation importance is that the latter permutes the feature of interest while keeping the correlated features constant.

Another widely used method is SHAP (SHapley Additive exPlanation) [19], which has theoretical foundations in Shapley values. They are a concept from cooperative game theory where the contribution of each player is given by the marginal contribution of the player to a coalition of players. It gives a local explanation for individual predictions, but aggregates are useful for assessing the GI of features. For example, the mean absolute Shapley values quantify the importance of the feature regardless of the direction of the impact on the prediction. SHAP values are an approximation of Shapley values, as the exact calculation of it is computationally expensive. There are different algorithms for approximation from which Kernel SHAP [19] is model-agnostic.

2.1.2 Tree-based model specific methods

The original classification and regression tree implementation as presented in [20] employed information gain as the criterion for splitting nodes. For regression trees, this approach aims to minimise loss through decisions made based on a given feature.

For ensemble tree models like random forest and gradient-boosted trees, a mean decrease in impurity (MDI) is used to quantify the importance of a feature. However, MDI tends to inflate the importance of continuous and high cardinality features [10]. The work [21] proposes split-based importance, which uses the number of splits of a feature as a measure of importance. The previously presented SHAP also has a tree-based solution for approximation called TreeSHAP [22], which bridges the gap between model-specific and agnostic methods. It has the advantage of computing in polynomial time instead of exponential time of the KernelSHAP method and addresses the issue of dependent features, such as correlated ones, by explicitly modelling the conditional expected prediction, thus avoiding extrapolation issues [10]. A disadvantage is that it can assign non-zero importance to the features that are not contributing to the prediction [10], however, in our case, it is not a problem as we are interested in the global feature importance.

2.2 Time series simulation

For evaluating the feature importance methods, real-world data is not suitable, as the ground truth of the importance of the features is not known; therefore, synthetic data is utilised to evaluate the feature importance methods. The work [23] describes two approaches for synthetic data generation. One relies on models that use real data to capture underlying distributions, and another one is based on existing models, for example, statistical models or background knowledge. Our study uses the latter approach as the properties of real data are not known, so synthetic data with known feature influences have to be generated.

2.3 Global feature importance challenges in multi-series forecasting

The work [15] points out multiple pitfalls concerning model interpretation in general, such as bad model generalisation and sensitivity to correlated features. In the following, we discuss the key challenges encountered when applying GFI, considering the specifics of the forecasting task:

2.3.1 Feature Transformations

In the SHAP documentation [24] is pointed out SHAP values can be squashed, by nonlinear feature transformations, which can lead to misleading results. Also, we observed that linear transformations can affect global multi-series models as scales of series are eliminated as inverse transformations are applied to the predictions.

2.3.2 Feature Dependency

PFI typically assumes feature independence and correlated features might cause PFI to underestimate the importance of features [15]. In forecasting tasks, typically the lagged values of the target variable are used as features that are usually correlated with the target and each other.

2.3.3 Extrapolation Sensitivity

PFI is sensitive to extrapolation, in which case the model is asked to make predictions outside the range of the training data. This is really important since some of the feature values are specifics of a certain series, and using unconditioned permutation might lead to invalid input. In this case, the PFI might not be reliable, as also the model might not be able to make accurate predictions outside the training range.

3 Methodology

3.1 Experiment design

To address the challenges, we designed a set of experiments to evaluate the effects of feature scaling, feature correlation, and extrapolation on the feature importance values.

The experimental setup is illustrated in Fig. 2 and includes the following experiment configurations: (i) Two data generation processes: linear and non-linear data generation process each designed to simulate different types of relationships,

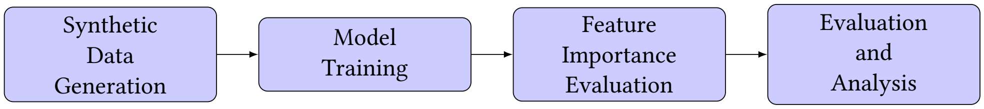


Figure 2: The experimental workflow design implemented in this study

(ii) Two categories of features: lagged variables and external demand drivers with additional noise; (iii) Data preprocessing with and without standardization to assess the impact of scaling on model performance and feature importance; (iv) Tree-based ensamble models Random Forest and LightGBM used to train the models; (v) Various feature importance methods: Tree gain and split importance, SHAP Tree explainer and PFI with grouped features; (vi) Evaluation of importance of both train and test datasets, to assess the model's generalization.

Additional configurations could be tested, for example, additional scaling methods, data generation processes, or models used.

3.2 Multi-series forecasting

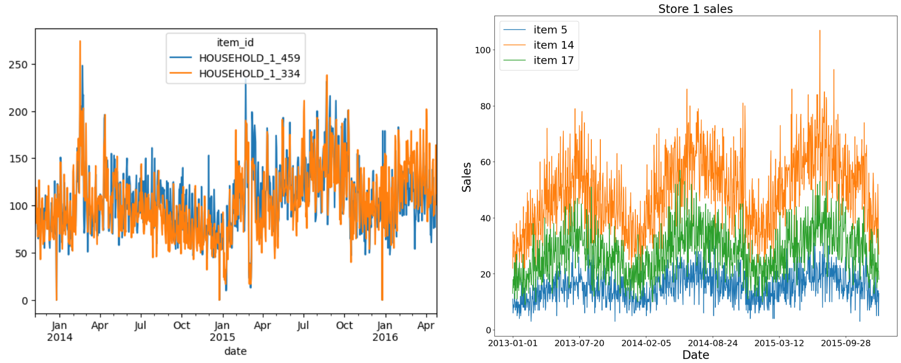
Multi-series modeling or global forecasting models are used to learn a single predictive model from multiple time series [25] This approach is beneficial when multiple time series share common patterns, which can be used to improve the generalization of models [26]. In demand forecasting, this is especially relevant when different products or point of sale exhibit similar seasonal trends or promotional effects. Other benefits include the easier evaluation of the model that has to be done on a single model and the easier maintenance.

3.3 Synthetic data generation

Given that the real-world datasets are often noisy and the ground truths regarding the feature contributions are not known. To address this issue, synthetic simulated datasets are used. The goal of the simulation is to create datasets with known dependencies between the features and target variables, which can be used for evaluation.

The reason for this is to imitate multiple real-world scenarios where the relationships between the features and the target variable are different. For instance, when examining the dataset from the M5 forecasting competition[1], various types of time series can be identified. Figure 3a illustrates a sample pair of household item series, where a distinct resemblance between them is noticeable. Items with consistent de-

[1] M5 Forecasting - Accuracy dataset: https://www.kaggle.com/c/m5-forecasting-accuracy/data



(a) M5 Forecasting - Accuracy dataset samples 1

(b) Store Item Demand Forecasting Challenge dataset samples 2

Figure 3: Example datasets for demand forecasting problems

mand, such as cleaning supplies and staple foods, generally exhibit similar trends. Other datasets like the one for *Store Item Demand Forecasting Challenge*[2] shown on the Fig. 3 have multiple series with different seasonal patterns of the target variable. Products such as ice-cream and sun cream are typically aligned with this seasonal trend.

Based on these example datasets, two data generation processes were developed: one linear and one non-linear. The linear data generation process consists of: i) Lag features: $x_{t-1}, \ldots, x_{t-n}$ are n lagged variables of the target variable, ii) External demand drivers: $ex_1, ex_2, \ldots, ex_n$ are external factors that influence the target variable, and iii) Noise: ϵ is the noise term. To obtain multiple time series which are related, we extend the linear data generation process to include a series-scaling factor s and a base demand b. By using the base demand and scaling factor we can initialize the data generation process for each series with different values. The scaling factor was not used on the lagged variables as they are already scaled by the target variable. The final formula for the linear data generation process is the following:

$$x_t = \beta_1 x_{t-1} + \ldots + \beta_n x_{t-n} + s \cdot (ex_1 + ..ex_n + b + \epsilon)$$

Similarly, the non-linear data generation consists of the following components. The basis of the data generation is a seasonal tren $S = A \cdot sin(P)$ where A denotes the amplitude and $P = (2 \cdot \pi)/365$ (365 days) is the period. To this we add a holiday effect H, a weekend effect W, a scaling factor $s_{1,2,3}$, and noise ϵ. The base demand b

[2] Store Item Demand Forecasting Challenge dataset: https://www.kaggle.com/competitions/demand-forecasting-kernels-only/data

is also included in the formula. The final formula for the non-linear data generation process is:

$$x_t = s \cdot S \cdot H \cdot W \cdot \epsilon \cdot b$$

3.4 Model training and evaluation

TODO: add more details Forecasting ML models are similarly trained on training and test dataset however the split and cross-validation strategy is different for time series data. The cross-validation strategy has to be time aware, meaning the order of the data has to be preserved. The folds are generated to include the previous data points in the training set and the next points in the test set, also known as rolling forecasting origin cross-validation [27].

To assess the model performance, multiple mertrics were used, including the coefficient of determination (R^2), mean absolute error(MAE), and mean squared error (MSE).

3.5 Global feature importance

We denote any machine learning model f that predicts the target variable y based on the input features $x = [x^1, x^2, ..., x^n]$. GFI of f is a measure of importance attributed to each $X = [X^1, X^2, ..., X^n]$ feature in overall. noted as $\phi(f, X) = [\phi(f, X^1), \phi(f, X^2), ..., \phi(f, X^n)]$. The model f is trained on the training set D_{train} and evaluated on the test set D_{test}.

For example, PFI measures the importance of a feature by shuffling the values of the feature and observing the change in the model's performance: $\phi(f, X_i)_{PFI} = L(f, D_{\text{test}}) - L(f, D_{\text{test}}^{X_i})$, where $D_{\text{test}}^{X_i}$ is the test set with the values of the feature X_i shuffled, and L is the loss function used to train the model. If MSE is used as the loss function, PFI is $\phi(f, X_i) = MSE(f, D_{\text{test}}) - MSE(f, D_{\text{test}}^{X_i})$.

To make the importance values comparable, we introduce the concept of relative importance, which is the ratio of the importance of a feature to the sum of all feature importances. The relative importance of a feature X_i is calculated as:

$$\text{RI}(f, X_i) = \frac{\phi(f, X_i)}{\sum_{j=1}^{n} \phi(f, X_j)}$$

Also the rank of the feature importance is calculated as the order of the feature importance values.

3.6 Addressing feature importance challenges

As presented in the previous section, multiple challenges can arise when calculating feature importance metrics, when working with time series data and tree-based models.

3.6.1 Feature scaling effect

Feature scaling is a common preprocessing step in machine learning to normalize the data. It is beneficial for models that are sensitive to sparse data or have different scales for the features. In the case of squashing transformations like log or square root, the feature importance values are also affected [24]. Linear transformations like normalization or standardization are used to scale the features to a common range. Fortunately, tree-based models are invariant to feature scaling, but scaling the features could be computationally beneficial. TODO: add references On the downside, by applying feature scaling the global (mean average) SHAP values will be scaled as well. To address this issue, we propose to apply the feature scaling on the data and compare the results with unscaled data.

3.6.2 Feature correlations

Kernel SHAP values assume that features are independent, which is not the case, especially in time series prediction models. Tree SHAP, however, can give more accurate results as the method considers joint feature distributions preserving dependencies [10]. Another approach that can be used to address the issue of correlated features is to cluster or group them. The work [28] introduces a grouped version of permutation importance, where the is to group the correlated features together and then permute the group of features. This grouping can be achieved through clustering; however, in case of time series, lag features may show strong autocorrelation. Grouping them together is logical while the external demand drivers can be used as separate features. This way it can be measured the amount of the importance is attributed to the lagged variables and how much to the external demand drivers.

3.6.3 Extrapolation

TreeSHAP can be used also to address the issue of extrapolation. Tree-path-dependent feature perturbation can be used as a solution, given that to calculate the SHAP values, there is no need for additional data points for sampling, giving an explanation true to the trained model [29]. This also makes it more computationally efficient too, compared to other Shapley methods that require data points to be sampled [10], [22].

4 Results

To evaluate different scenarios on the simulated datasets, multiple experiments were conducted with LightGBM and Random Forest models to evaluate the feature importance methods. The main implementations of the FI method were the Permutation Feature Importance (PFI) of sklearn[3], the TreeExplainer SHAP from fasttreeshap[4], and the built-in Tree Gain and Split Importance from LightGBM[5], with only the Tree Gain used for Random Forest[6].

4.1 Synthetic data generation

For the linear data generation process, two lagged variables with $\beta_{1,2} = [0.6, 0.2]$ weights were used. As exogenous variables, two additional features were generated, one representing a holiday effect present or not given by a binary random variable, and an additional external factor called temperature with a normal distribution $N(20, 5)$. Over the result, scaling factors $s_{1,2,3} = [0.5, 1, 2]$ were applied, with base demand $b = 100$ and noise $\epsilon = \mathcal{N}(1, 0.05)$. The concrete example is shown in Fig. 4a. Inspecting the partial autocorrelation of the time series in Fig. 4b it can be observed that the first two lags have the highest value, meaning that the first two lags explain most of the variance.

To generate the dataset, with the non-linear feature contributions, an additional time series generator library[7] was used. The following components were used to generate the data, scaling $s_{1,2,3} = [0.5, 1, 2]$, base demand $b = 100$, noise $\epsilon = \mathcal{N}(1, 0.05)$, holiday effect $H = 1.5$ with fade-in and out effect and seasonality $S = A * sin(P)$ where $A_{1,2,3} = [0.2, 0.3, 0.4]$ and weekend effect $W = 1.3$ if weekend, 1 otherwise. The concrete example is shown in Fig. 5b.

4.2 Model training and evaluation

Several LightGBM and Random Forest models were trained on the generated datasets with and without scaling. To obtain predictive models with similar performance, the optuna library [8] was used for parameter search with only the number of estimators

[3] sklearn - Permutation Feature Importance: https://scikit-learn.org/stable/modules/permutation_importance.html

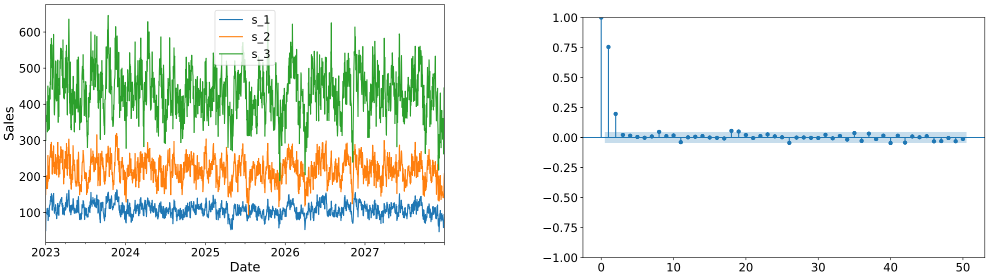
[4] fasttreeshap - TreeExplainer: https://github.com/linkedin/FastTreeSHAP

[5] LightGBM - Feature Importance:https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html

[6] Random Forest - Feature Importance: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[7] Time Series Generator - http://https://github.com/Nike-Inc/timeseries-generator

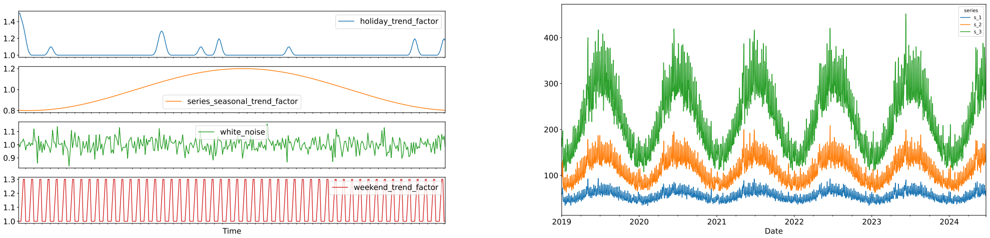
[8] Optuna: https://optuna.org



(a) Linear data generation process example of 3 series with scaling $s_{1,2,3} = [0.5, 1, 2]$, base demand $b = 100$

(b) Partial autocorrelation of lagged variable for series 1

Figure 4: Linear data generation process



(a) Components of the non-linear data generation process

(b) Non-linear data generation process of 3 series with scaling $s_{1,2,3} = [0.5, 1, 2]$, base demand $b = 100$

Figure 5: Non-linear data generation process

as a hyperparameter.

The results of the model training are shown in Table 1 and Table 2. The difference between the performance of the models with and without scaling is not significant, but the number of estimators is different.

4.3 Feature Importance Metrics Comparison

For the evaluation the above mentioned feature importance methods were used. In addition, for some of the methods in which a dataset was expected for calculation, both the train and the test datasets were used separately. To evaluate the similarity of the feature importance values, the Spearman correlation coefficient was used. Given the high number of experiments, the results for the LightGBM model are included in Fig. 6 for the linear dataset. With standardization, the consistency

Table 1: Test results for LightGBM and Random Forest models on linear dataset

Model	Dataset	Estimators	R^2	MAE
LightGBM	Scaled	349	0.9775	15.058390
LightGBM	Non-Scaled	415	0.9774	15.187829
Random Forest	Scaled	1179	0.9789	14.504917
Random Forest	Non-Scaled	565	0.9791	14.450996

Table 2: Test results for LightGBM and Random Forest models on non-linear dataset

Model	Dataset	Estimators	R^2	MAE
LightGBM	Scaled	349	0.9895	5.734598
LightGBM	Non-Scaled	1026	0.9871	6.289761
Random Forest	Scaled	897	0.9898	5.663916
Random Forest	Non-Scaled	225	0.9877	6.261677

of importance values can be enhanced across the different estimation methods, as demonstrated in Subfig. 6a, while the correlation plot on Subfig. 6b for the model trained on non-standardized dataset emphasizes that without standardization tree specific importance values are less or not correlated with the other methods. In both cases *TREE_SPLIT* importance is highlighted as the least correlated, the reason being that the model prefers splits on the continuous variables, as shown in Table 3. The main reason can be observed in Table 3, where the relative importance values show that the model splits heavily on continuous variables, compared to the rest of features, even thogh the third and fourth lags have low average contribution by SHAP values and low PFI.

The PFI and SHAP values are more similar, only minor differences are present due to the usage of the test or train dataset in calculation. Table 3 illustrates the relative importance of the features for the linear dataset, allowing for a clearer comprehension.

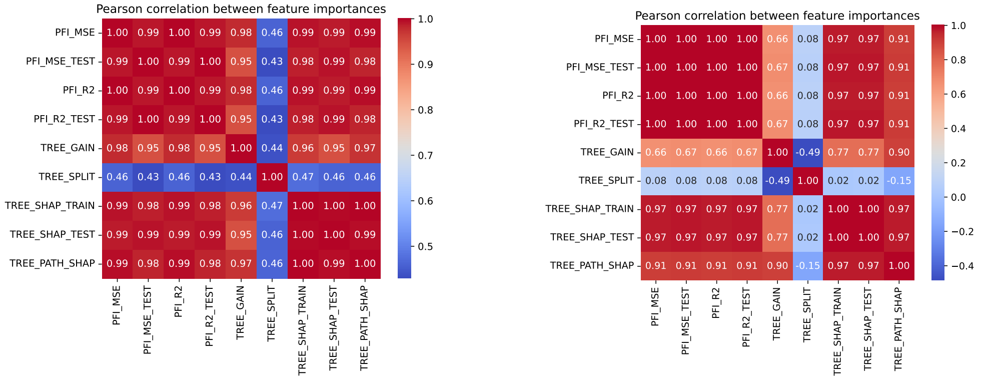
4.4 Feature scaling effect on importance measures

Feature scaling affects the models this way, the importance values are shifted to other features as show in 3. Without standardization, the models rely heavily on the ‘$_{l}evel'feature,whichistheidentifierofthetimeseries,butthe‘TREE_{S}PLIT‘valuesarelowsignalingthatonly$

The violin plots of SHAP values for a single series, shown in Figure 7 for the LightGBM model and Figure 8 for the Random Forest model, illustrate how feature importance values differ between standardized and non-standardized datasets. Without

Table 3: Relative importance of features for the LightGBM model with the linear dataset

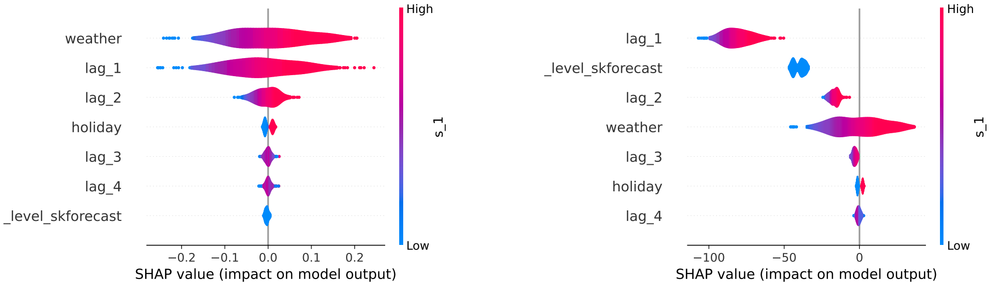
	Method	lag_1	weather	lag_2	lag_4	lag_3	holiday	_level
Non-Standardized	PFI_MSE	62.14	7.79	3.15	0.43	0.47	0.17	25.85
	PFI_MSE	62.14	7.79	3.15	0.43	0.47	0.17	25.85
	PFI_MSE_TEST	62.58	8.01	2.58	0.10	0.12	0.11	26.52
	PFI_R2	62.14	7.79	3.15	0.43	0.47	0.17	25.85
	PFI_R2_TEST	62.58	8.01	2.58	0.10	0.12	0.11	26.52
	TREE_GAIN	31.62	3.51	1.79	0.13	1.87	0.06	61.02
	TREE_PATH_SHAP	38.51	12.57	6.78	1.25	3.44	1.66	35.79
	TREE_SHAP_TEST	43.96	13.52	8.94	1.50	2.08	1.71	28.29
	TREE_SHAP_TRAIN	43.89	13.17	9.20	1.66	2.11	1.58	28.39
	TREE_SPLIT	20.51	21.77	18.75	17.57	18.33	2.62	0.45
Standardized	PFI_MSE	47.04	41.86	6.60	1.63	1.61	0.74	0.52
	PFI_MSE_TEST	45.29	49.90	4.22	−0.01	0.00	0.58	0.01
	PFI_R2	47.04	41.86	6.60	1.63	1.61	0.74	0.52
	PFI_R2_TEST	45.29	49.90	4.22	−0.01	0.00	0.58	0.01
	TREE_GAIN	55.76	35.05	6.16	1.19	1.15	0.49	0.20
	TREE_PATH_SHAP	40.32	36.69	11.50	2.51	2.88	4.58	1.53
	TREE_SHAP_TEST	37.33	39.72	11.57	2.43	2.65	4.60	1.70
	TREE_SHAP_TRAIN	39.10	37.00	12.23	2.74	2.90	4.45	1.57
	TREE_SPLIT	18.93	19.22	18.83	19.72	18.01	2.52	2.77



(a) With standardization

(b) Without standardization

Figure 6: Correlation of feature importance values for LightGBM model



(a) Standardized

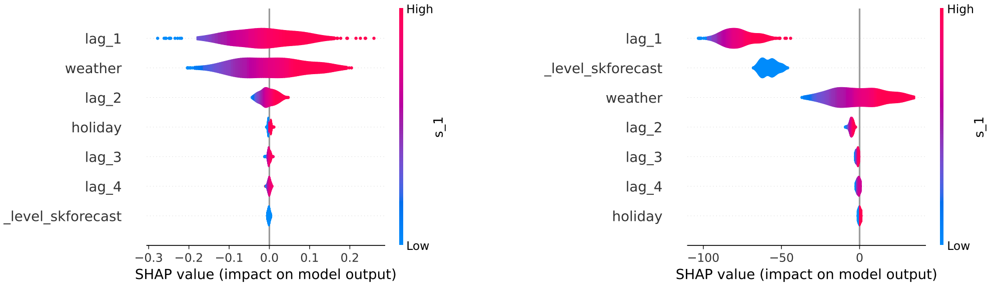
(b) Non-standardized

Figure 7: Feature importance values for LightGBM model for series 1

scaling the ‘$_level‘valueshavemuchlargermagnitude, evendominatingasincaseoftheRandomForestmodelf$

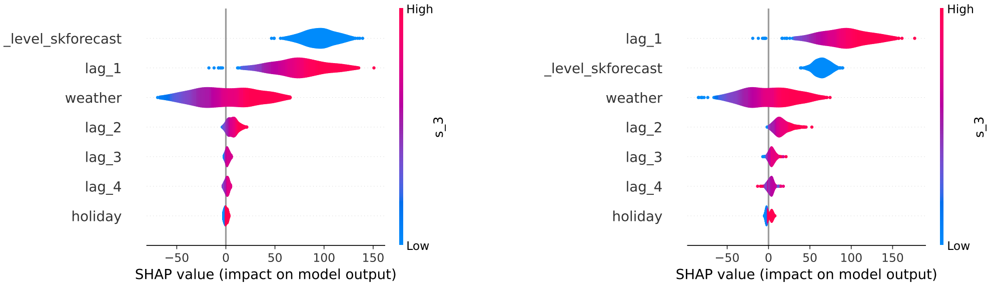
4.5 Correlation of features and extrapolation

In Fig. 10 the grouped PFI by the coefficient of determination for the Random Forest model with standardisation is shown. As expected, the importance of the grouped lag variables is the highest,followed by the exogenous variables. The relative importances are shown in Table 4 for linear and Table 5 for non-linear datasets. In the case of the linear dataset by scaling, the importance of the lag values are reduced, but the relative importance for the two models is similar. The value of approximately 0.6 for the lag variable makes sense, as the ‘lag_1‘ contributed 0.6 to the target in data generation and the ‘lag_2‘ variable has its contribution reduced by the ‘lag_1‘



(a) Standardized (b) Non-standardized

Figure 8: Feature importance values for Random Forest model for series 1



(a) Random forest (b) LightGBM

Figure 9: Feature importance values for Random Forest and LightGBM models without standardization for series 3

Table 4: Relative feature importance for grouped PFI for the linear combined dataset.

	lgbm_scaled	lgbm_nonscaled	rf_scaled	rf_nonscaled
_level_skforecast	0.000103	0.196046	-0.000194	0.439479
holiday	0.004765	0.000798	0.002362	0.000293
lag_	0.587014	0.743967	0.592377	0.501273
weather	0.408118	0.059189	0.405455	0.058956

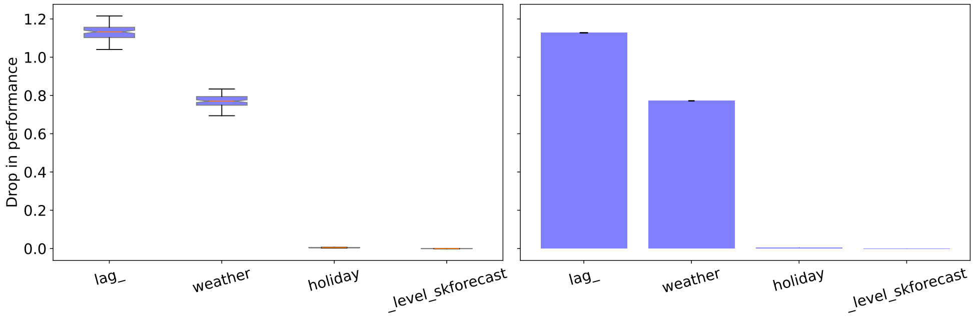


Figure 10: Grouped permutation feature importance Random Forest model with standardization

Table 5: Relative feature importance for grouped PFI for the non-linear combined dataset.

	lgbm_scaled	lgbm_nonscaled	rf_scaled	rf_nonscaled
_level_skforecast	0.018005	0.029358	0.011419	0.010066
day_of_year	0.198566	0.021532	0.070868	0.006378
holiday_trend_factor	0.029211	0.004774	0.016281	0.002154
lag_	0.381743	0.868834	0.571649	0.918375
weekend_trend_factor	0.372476	0.075502	0.329783	0.063028

variable.

The results of the non-linear dataset in Table 5, the importance of lag variables is the highest showing that the model relies almost solely on the lagged values to make predictions, highlighting the fact of the importance of data scaling.

5 Discussion

For practitioners, it is important to understand the implications of these methods when interpreting the results. In case of feature scaling, by not applying scaling to the features, the importance values might seem more intuitive. But the consequence of this is that the importance of features might be shifted to other features, as in our case, the identifier of the time series.

If using permutation importance, it is important to have in mind that correlated features can unexpected results. We recommend using grouped importance to get a value closer to the real importance values while also partially addressing extrapolation issues.

SHAP values can be really useful, especially without scaling as they provide not only importance but also the contribution of feature to prediction. It remains to be evaluated whether applying the inverse transformations on the scaled SHAP results can provide better results. Tree SHAP with perturbation method based on the fitted tree can result in a faster result, which can be useful in the case of large datasets while also addressing some concerns with regard to extrapolation. Using the perturbation method on Tree SHAP applied to the trained tree can lead to faster outcomes, which is advantageous for large data sets, while also mitigating some extrapolation issues. Examining the values at the series level may provide further insights into the directional contributions of the features. Given the complexity of the problem, we recommend using multiple methods to better understand the model and features.

The data generation process does not address all real-world complexities. As a first step, we only considered independent time series, but in other demand forecasting scenarios the sales of one product might influence the sales of another product.

With regard to the application of SHAP values, there are some limitations. Although some methods like LightGBM can handle categorical features, SHAP libraries do not support them. Scaling the features also might result in spurious errors due to the precision of the floating point numbers. Split- and gain-based importance for trees might be easy to calculate, but they are not always the best choice for evaluating feature importance. For causal analysis, feature importance methods are not enough, but they can provide a good starting point.

Our study demonstrates the use of post-hoc feature importance methods in multi-series forecasting. It was shown that the choice of the feature importance method can have a significant impact on the results. In addition to this, it was also presented that not only squashing transformations can result in different importance results, but linear transformations can affect the outcome due to the different model behavior. Last but not least some of the caveats of the calculation of feature importance values was also discussed and addressed.

6 Conclusion

There are several pitfalls while evaluating feature importance of tree-based models in the context of multi-series forecasting. Our goal was to point out some of these pitfalls and find ways to mitigate them. We have shown that the choice of feature importance method can have a significant impact on the results. In addition, when calculating the FI values, scaling of features can also have an impact on the results.

The novelty of our work lies in multiple aspects. First, by developing and using a simulation framework for benchmarking feature importance methods, we aim to provide reproducible results and comparability of different methods. In addition, we identify common issues in the evaluation of the importance of features in demand forecasting models. These findings are relevant not only for the field of multi-series demand forecasting but also for other times series forecasting tasks.

Future work could involve more complex data generation processes, such as correlated time series, or the study of other data structures, such as hierarchical time series for demand forecasting. Inverse scaling of feature contributions of SHAP given the scaling coefficients of the features could also be a potential practical research direction.

Our work contributes towards the development of trustworthy AI-driven decision support systems in demand forecasting, by providing insights into the post-hoc evaluation of tree-based models in multi-series forecasting.

Data Availability: The source code and synthetic dataset used in this study are available on GitHub. [9] The datasets used and generated during this study are publicly available at ¡repository name¿ and can be accessed at ⟨dataset_url⟩. These datasets include simulated multi-series forecasting data and feature importance metrics.

Acknowledgments: This section is designated for expressing gratitude for any assistance received that does not fall within the scope of the author contribution or funding acknowledgments. Here, you may acknowledge support from individuals, organizations, or institutions that contributed indirectly to the research or manuscript preparation, such as administrative support, technical assistance, critical reviews, or informal advice.

References

[1] F. Lampathaki, E. Bosani, E. Biliri, *et al.*, "XAI for Product Demand Planning: Models, Experiences, and Lessons Learnt," in *Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through AI*, J. Soldatos, Ed. Cham: Springer Nature Switzerland, 2024, pp. 437–458, ISBN: 978-3-031-46452-2. DOI: 10.1007/978-3-031-46452-2_25. [Online]. Available: https://doi.org/10.1007/978-3-031-46452-2_25 (⇒ 2, 3).

[9] Synthetic dataset https://github.com/mkk-phd-work/feature_importance_challenges.git.

[2] M. A. Mediavilla, F. Dietrich, and D. Palm, "Review and analysis of artificial intelligence methods for demand forecasting in supply chain management," *Procedia CIRP*, vol. 107, pp. 1126–1131, 2022, Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022, ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2022.05.119. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827122004036 (⇒ 2).

[3] B.-E.-M. Mursa, M. Kuti-Kreszács, C. Moroz-Dubenco, and F. Bota, "Facilitating model training with automated techniques," *Studia Universitatis Babeș-Bolyai Informatica*, vol. 68, no. 2, pp. 53–68, 2023. DOI: 10.24193/subbi.2023.2.04. [Online]. Available: https://studia.reviste.ubbcluj.ro/index.php/subbinformatica/article/view/7007 (⇒ 2).

[4] Y. Du, A. Rafferty, F. Mcauliffe, L. Wei, and C. Mooney, "An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus," *Scientific Reports*, vol. 12, Jan. 2022. DOI: 10.1038/s41598-022-05112-2 (⇒ 3).

[5] J. Černevičienė and A. Kabašinskas, "Explainable artificial intelligence (xai) in finance: A systematic literature review," *Artificial Intelligence Review*, vol. 57, no. 8, p. 216, Jul. 2024, ISSN: 1573-7462. DOI: 10.1007/s10462-024-10854-8. [Online]. Available: https://doi.org/10.1007/s10462-024-10854-8 (⇒ 3).

[6] A. Kuznietsov, B. Gyevnar, C. Wang, S. Peters, and S. Albrecht, *Explainable ai for safe and trustworthy autonomous driving: A systematic review*, Feb. 2024. DOI: 10.48550/arXiv.2402.10086 (⇒ 3).

[7] *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts*, Official Journal of the European Union, L 327, 12 July 2024, pp. 1-60, Accessed: 18 December 2024, 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj (⇒ 3).

[8] L. Tronchin, E. Cordelli, L. R. Celsi, *et al.*, "Translating Image XAI to Multivariate Time Series," *IEEE Access*, vol. 12, pp. 27 484–27 500, 2024. DOI: 10.1109/ACCESS.2024.3366994 (⇒ 3).

[9] R. Assaf and A. Schumann, "Explainable deep neural networks for multivariate time series predictions," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, International Joint Conferences on Artificial Intelligence Organization, Jul. 2019, pp. 6488–6490. DOI:

10.24963/ijcai.2019/932. [Online]. Available: https://doi.org/10.24963/ijcai.2019/932 (⇒ 3).

[10] C. Molnar, *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*, 2nd ed. online, 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book (⇒ 3–5, 10).

[11] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018, ISSN: 0360-0300. DOI: 10.1145/3236009. [Online]. Available: https://doi.org/10.1145/3236009 (⇒ 3).

[12] H. Mandler and B. Weigand, "A review and benchmark of feature importance methods for neural networks," *ACM Comput. Surv.*, vol. 56, no. 12, Oct. 2024, ISSN: 0360-0300. DOI: 10.1145/3679012. [Online]. Available: https://doi.org/10.1145/3679012 (⇒ 4).

[13] N. Agarwal and S. Das, "Interpretable machine learning tools: A survey," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 1528–1534. DOI: 10.1109/SSCI47803.2020.9308260 (⇒ 4).

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324 (⇒ 4).

[15] C. Molnar, S. Gruber, and P. Kopper, *Limitations of interpretable machine learning methods*, 2020. [Online]. Available: https://slds-lmu.github.io/iml_methods_limitations/ (⇒ 4, 6).

[16] C. Molnar, C. Molnar, G. König, *et al.*, "Pitfalls to avoid when interpreting machine learning models," *arXiv.org*, 2020. DOI: null (⇒ 4).

[17] G. Hooker, L. Mentch, and S. Zhou, "Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance," *Statistics and Computing*, vol. 31, no. 6, pp. 1–16, Nov. 2021. DOI: 10.1007/s11222-021-10057-z (⇒ 4).

[18] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, p. 307, Jul. 2008, ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-307. [Online]. Available: https://doi.org/10.1186/1471-2105-9-307 (⇒ 4).

[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf (⇒ 4, 5).

[20] L. Breiman, *Classification and regression trees*. Routledge, 2017 (⇒ 5).

[21] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *arXiv: Learning*, 2016. DOI: 10.1145/2939672.2939785 (⇒ 5).

[22] S. M. Lundberg, G. Erion, H. Chen, *et al.*, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, DOI: 10.1038/s42256-019-0138-9. [Online]. Available: https://par.nsf.gov/biblio/10167481 (⇒ 5, 10).

[23] K. E. Emam, L. Mosquera, and R. Hoptroff, *Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data*. O'Reilly Media, Incorporated, 2020, ISBN: 9781492072744 (⇒ 5).

[24] *How a squashing function can effect feature importance*, SHAP documentation. [Online]. Available: https://shap.readthedocs.io/en/latest/example_notebooks/tabular_examples/model_agnostic/Squashing%20Effect.html (⇒ 6, 10).

[25] T. Januschowski, J. Gasthaus, Y. Wang, *et al.*, "Criteria for classifying forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 167–177, 2020 (⇒ 7).

[26] A. Buonanno, M. Caliano, A. Pontecorvo, G. Sforza, M. Valenti, and G. Graditi, "Global vs. local models for short-term electricity demand prediction in a residential/lodging scenario," *Energies*, vol. 15, no. 6, 2022, ISSN: 1996-1073. DOI: 10.3390/en15062037. [Online]. Available: https://www.mdpi.com/1996-1073/15/6/2037 (⇒ 7).

[27] R. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice 3rd ed," *O Text*, 2018 (⇒ 9).

[28] L. Plagwitz, A. Brenner, M. Fujarski, and J. Varghese, "Supporting AI-Explainability by Analyzing Feature Subsets in a Machine Learning Model," in May 2022, vol. 294, ISBN: 9781643682846. DOI: 10.3233/SHTI220406 (⇒ 10).

[29] H. Chen, J. D. Janizek, S. M. Lundberg, and S. Lee, "True to the model or true to the data?" *CoRR*, vol. abs/2006.16234, 2020. arXiv: 2006.16234. [Online]. Available: https://arxiv.org/abs/2006.16234 (⇒ 10).

Received: DD MM 20yy; *Revised: DD MM 20yy*; *Accepted: DD MM 20yy*