COMMITTEE ON DATA
CODATA
INTERNATIONAL SCIENCE COUNCIL

A Notion of Feature Importance by Decorrelation and Detection of Trends by Random Forest Regression

RESEARCH PAPER

]u[ubiquity press

YANNICK GERSTORFER
MAX HAHN-KLIMROTH
LENA KRIEG

*Author affiliations can be found in the back matter of this article

ABSTRACT

In many studies, we want to determine the influence of certain features on a dependent variable. More specifically, we are interested in the strength of the influence – i.e., is the feature relevant? And, if so, how the feature influences the dependent variable. Recently, data-driven approaches such as *random forest regression* have found their way into applications (Boulesteix et al. 2012). These models allow researchers to directly derive measures of feature importance, which are a natural indicator of the strength of the influence. For the relevant features, the correlation or rank correlation between the feature and the dependent variable has typically been used to determine the nature of the influence. More recent methods, some of which can also measure interactions between features, are based on a modeling approach. In particular, when machine learning models are used, SHAP scores are a recent and prominent method to determine these trends (Lundberg et al. 2017).

In this paper, we introduce a novel notion of feature importance based on the well-studied Gram-Schmidt decorrelation method. Furthermore, we propose two estimators for identifying trends in the data using random forest regression, the so-called absolute and relative traversal rate. We empirically compare the properties of our estimators with those of well-established estimators on a variety of synthetic and real-world datasets.

CORRESPONDING AUTHOR:
Yannick Gerstorfer
Frankfurt Institute for Advanced Studies, Goethe University Frankfurt Frankfurt 60325, Germany
gerstorfer@fias.uni-frankfurt.de

KEYWORDS:
Data Science; Explainable Artificial Intelligence; Feature Importance; Gram-Schmidt Decorrelation; Random Forest Regression; Shapley Value; Trend Estimation

TO CITE THIS ARTICLE:
Gerstorfer, Y, Hahn-Klimroth, M and Krieg, L. 2023. A Notion of Feature Importance by Decorrelation and Detection of Trends by Random Forest Regression. *Data Science Journal*, 22: 42, pp. 1–14. DOI: https://doi.org/10.5334/dsj-2023-042

1. INTRODUCTION

In many studies, scientific researchers are faced with high-dimensional but limited data to determine the influence of specific features on a dependent variable. Typically, the data consist of both numerical and categorical features, and strong artificial multivariate correlations appear. In particular, when data are generated from observations of live animals or collected in medical procedures, it is very likely that the data are unbalanced and, even worse, not all combinations of features contain samples. Therefore, it is unlikely that all necessary assumptions of classical statistical tests will be met. Machine learning methods have gained popularity among researchers because they can produce robust effect estimates with minimal assumptions. However, in particular, plain machine learning models are prone to overfitting effects such that they need to be applied with care. A plain but prominent example is the *random forest regression*. While random forests are rather old concepts in the mathematics literature, due to advances in data science concepts as well as the increasing computational power available to any research group, they have been finding their way into life science studies rather recently (Boulesteix et al. 2012; Gübert et al. 2023). Random forest regression, as with all machine learning models, makes few assumptions about the distributions of the underlying data and is particularly robust to noise and outliers. Finally, it allows to directly derive measures of feature importance, which are a natural indicator of the strength of influence of individual features (Fraser 1965; Beraha et al. 2019; Parveen et al. 2012). In cases where classical statistical tools such as ANOVA can be applied, it is well known that most features found to be significant by ANOVA also have high feature importance and vice versa (Chicco & Jurman 2021; Saarela & Jauhiainen 2021).

Once relevant features have been found, it is important to determine how the values of the features affect the dependent variable. Probably the oldest approach is to measure the correlation or rank correlation between a feature and the dependent variable. More recent methods, some of which can also measure interactions between features, are based on a modeling approach. A model (e.g., a multivariate linear regression model) is trained and its parameters can be used to determine trends, especially when machine learning models are used, the SHAP scores (Shapley 1953) are a recent and prominent method to determine these trends. These approaches use the model rather than the raw data. This can help to identify trends that are not directly visible in the data but are hidden behind noise. On the other hand, a decent model is required so that these trends are reliable.

The goal of this paper is twofold. First, since dependencies between features are known to influence feature importance scores, we introduce a notion of feature importance based on the well-studied Gram-Schmidt decorrelation method. This notion is empirically compared with a similar approach based on residual learning and the classical impurity-based feature importance and permutation importance. Second, we propose two estimators to identify trends in the data using random forest regression. We exploit the structure of random forests, i.e., at each split node we can compare the average prediction in the left and right subtrees. Since the left subtree is built on data below a threshold and the right subtree contains data above that threshold, this induces a natural estimator of a joint trend between the feature and the predicted variable.

2. BACKGROUND AND NOTATION

2.1 FEATURE IMPORTANCE

With respect to random forests, three types of feature importance scores are well known in the literature. The first one is an *impurity-based* feature importance. The so-called impurity is quantified by the splitting criterion of the collection of contained decision trees. Therefore, it is likely to overestimate the importance of large numerical features (if the dataset is not standardized). Furthermore, it is possible that features that may not be predictive on unseen data are found to be important in the case of overfitting. For these reasons, a second type of feature importance, the so-called *permutation importance*, has found its way into the literature and is to be preferred. It is defined as the decrease in model performance when a single feature's values are randomly shuffled. A similar possibility to measure a feature's importance is based on *exclusion* of a variable. More precisely, to measure the importance of a feature, a second random forest is trained on the same data excluding the feature under consideration. The importance is, again, measured as the decrease in the model's performance. Of course, the permutation-based and the exclusion-based approach have their shortcomings if there are clusters of (highly) correlated features (Breiman 2001). Indeed, due to the availability of the modified feature's values via the correlation, the importance score can be highly misleading. One approach to overcome this problem, which is often used in the process of feature extraction, is to keep only one variable per cluster (Chen et al. 2020; Louppe 2014; Guyon et al. 2002). If the ultimate goal is to design a decent prediction model with as few features as possible, this is the state of the art. But in some cases, researchers are actually more interested in estimating the importance of each feature to determine which features influence the dependent variable and how strongly. In this setting, it may be convenient to treat the correlations differently. There are at least two *decorrelation techniques* that are usually used either for clustering data or for designing well-performing prediction models: the Gram-Schmidt decorrelation technique (Zhang & Chan 2006) or residual-based decorrelation (Dezfouli et al. 2019). The main idea in both cases is to subtract the information from a given feature F_i given by F_1,..., $F_{i\text{-}1}$, F_{i+1},... F_d and use this *residual* to train the model.

2.2 TRENDS

We compare three different ways to define *trends* in the dataset. The simplest way one might think of examining a trend between the values of a feature Y and the predicted variable is to use the *correlation coefficient* $r(X,Y) = \mathrm{Cov}(X,Y)/(\sigma(X)\sigma(Y))$, which reflects linear trends. A more general correlation coefficient that handles any *monotone* trends are various types of rank correlation coefficients such as the *Spearman correlation coefficient* $\rho(X,Y) = r(R(X), R(Y))$ where $R(\cdot)$ denotes the rank function. This method of finding trends is well established and only considers the observable raw data.

Another approach does not look at the raw data but fits a model and looks for trends in that model. Many practitioners tend to identify trends in multivariate tasks by fitting a linear model to the data and interpreting the sign and corresponding p-value of the coefficient of a feature as a trend. We will denote this coefficient by $r_{LM}(F)$. However, we will see that this can be very misleading, even for very simple datasets.

In recent years, an old concept from mathematical game theory, called *Shapley values*, has been used to interpret machine learning models (Lundberg & Lee 2017). In particular, they are well understood mathematically for tree-based models and random forests. The Shapley value of a feature with respect to a data point measures how much the feature value contributes to the prediction compared to the average prediction and is defined as the average marginal contribution of the feature value among all possible combinations of features. For a formal definition, see Shapley's original paper (1953), and for a detailed discussion of how to use the concept in machine learning, see Janzing et al. (2020) and Sundararajan and Najmi (2020). Shapley scores can be used to determine trends. More precisely, if a low value of a feature X induces a decrease of the Shapley value, then the model predicts a negative trend of the predictive variable with respect to feature X.

2.3 STUDIED DATASETS

To test the performance of our estimators in practice, we use two very well-known real datasets, called *Kaggle fish market dataset* (FISH) (Pyae 2019) and *California housing data* (HOUSING) (Nugent 2017). In addition, we create multiple different synthetic datasets to explore certain aspects of the estimators.

FISH contains the records of seven different common fish species in fish market sales. The features are species, weight, vertical length, diagonal length, transverse length, height, and width for each fish. Of these characteristics, we used weight, height, and width to predict vertical length. The California housing data refers to the houses found in a given California county and summary statistics based on 1990 census data. The features are longitude, latitude, median age of the house, total number of rooms, total number of bedrooms, population, number of households, median income, and ocean proximity for each county with median house value as the prediction target. We transformed the ocean proximity feature into an ordinal scale.

The first synthetic datasets (SYN1(a) and SYN1(b)) are derived from a base dataset *B* consisting of 1000 samples and 10 features, three of which are informative. The base dataset is standardized by removing the mean and scaling to unit variance. It is then combined with a Gaussian noise dataset *N* standardized the same way and with the same structure but no informative features, resulting in dataset SYN1(a). For SYN1(b), the same procedure is applied using uniform white noise rather than Gaussian noise. A family of datasets is obtained by:

$$D_w = (1-w)B + wN \quad (w \in [0.01, 0.02, \ldots, 1.]).$$

SYN1(a) and SYN1(b) each consist of the combination of the base dataset with 250 different random noise datasets. These datasets are used to compare the robustness of trend estimators.

The second synthetic dataset (SYN2(a)) consists of 100 samples with independently generated features:

$$X_0 \sim 3 \cdot \mathcal{N}(0,1), \qquad X_1 \sim 2 \cdot \mathcal{N}(0,1), \qquad X_2 \sim \mathcal{N}(0,1).$$

Furthermore, given X_0, we define:

$$A_0 = X_0 + \mathcal{N}(0,1), \qquad A_1 = X_0 + 10 \cdot \mathcal{N}(0,1), \qquad A_2 = X_0^2 + 10 \cdot \mathcal{N}(0,1).$$

The true label is given by:

$$Y = 4X_0^{1.5} + 2X_1 + 0.5X_2^2.$$

Thus, the real labels depend on X_0, X_1, X_2 and A_0, A_1, A_2 can be considered as noisy instances of X_0 with different types of dependencies. Similarly, we define a set SYN2(b), in which the standard Gaussian $\mathcal{N}(0, 1)$ is replaced by a uniform [–1,1] random variable. This is an instance of *white noise* applied to data. SYN2(a) and SYN2(b) are used to compare different notions of feature importance.

The last synthetic datasets (SYN3(a and b)) consist of 100 samples with only one informative feature X_0, defined as previously. Moreover, A_0, A_1, A_2 are defined as above. The true labels are now given by $Y = 4X_0^{1.5}$. SYN3 (a and b) are used to compare the notions of feature importance on a *cluster* of correlated features, in direct comparison to SYN2, in which two additional, uncorrelated, informative features are present.

3. CONTRIBUTION

3.1 FINDING TRENDS IN A DATASET

We compare the commonly used regression coefficients *r*, *ρ*, the linear model-based trend estimator, a Shapley-based trend estimator, and propose two novel estimators based on random forest regression to determine the trends of features. For this purpose, we simply define the Shapley-based trend of a feature as the correlation between its values *X* and its Shapley values *s*(*X*), so that we obtain the estimators *r*(*X*, *s*(*X*)) and *ρ*(*X*, *s*(*X*)), respectively. *Moreover, we report the Shapley values themselves visually to compare the actual marginal contributions to the corresponding trend estimators.*

The two proposed trend estimators are the absolute and the relative *traversal rate*. The random forest regression model uses an ensemble of uncorrelated decision trees. At each node, the current dataset is partitioned into two partition classes based on the values of the node's feature. We assume without loss of generality that the data in the *left* partition class belong to small feature values and the data in the *right* partition class belong to large feature values. To determine the trend of a feature *F*, we compare the mean of the features in the left and right partition classes per node. If the average value of the predicted variable in the left tree is smaller than in the right tree, this corresponds to a positive correlation with the feature *F*. The intuition behind this is the following. As we find only data points with a smaller value of *F* in the left tree than in the right tree, and the model predicts a smaller value of the predicted variable, we expect a positive correlation between the feature *F* and the dependent variable. More formally, let $\{F_j\}_{j=1\ldots n}$ denote the set of nodes in the random forest in which the data is partitioned with respect to feature *F*. The corresponding partition classes are called $L(F_n)$ and $R(F_n)$. If the feature

F is clear from the context, we abbreviate these classes to L_n and R_n. Furthermore, for a subset A of the values of the predicted variable, we define $AVG(A) = |A|^{-1}\sum_{a\in A} a$ as the average value of the set A. This allows us to define our trend estimators.

Definition 1

Given a random forest $\mathcal{R}$, let $\{F_j\}_{j=1\ldots n}$ denote the set of nodes in the random forest in which the data is split with respect to feature F. The absolute traversal rate of feature F is defined as:

$$ATR(\mathcal{R},F) = n^{-1}\sum_{i=1}^{n}\left(1\{AVG(L(F_i)) \le AVG(R(F_i))\} - 1\{AVG(L(F_i)) \ge AVG(R(F_i))\}\right).$$

Moreover, the relative traversal rate of feature F is defined as:

$$RTR(\mathcal{R},F) = 2\sum_{i=1}^{n}\frac{R(F_i)-L(F_i)}{|L(F_i)+R(F_i)|}.$$

The ATR formalizes the idea that we have a trend when the feature with a higher value causes the model to return a higher value. It is a number between –1 and 1. If it takes a value close to 1, this means that in almost every split, the smaller feature's values induce a smaller model prediction, while a value close to –1 means that in most splits the smaller feature's values induce a larger model prediction. A value around zero means that about half of the splits induce a positive, and half of the splits a negative correlation.

The RTR is a weighted variant of this idea. If the relative difference between the average prediction in the left and right sub-tree is large, the split can be said to induce a stronger trend than in the case in which the average predictions are close to each other in both trees.

Both ATR and RTR estimate the impact of a feature by assessing how it splits the tree into partition classes (see Figure 1). In particular, the impact of a feature is estimated by comparing those partition classes, which can be done without knowing any of the other features. This is in contrast to the Shapley-based trend estimator, which assesses the impact of a feature by calculating the average marginal contribution of a feature over different coalitions of features.

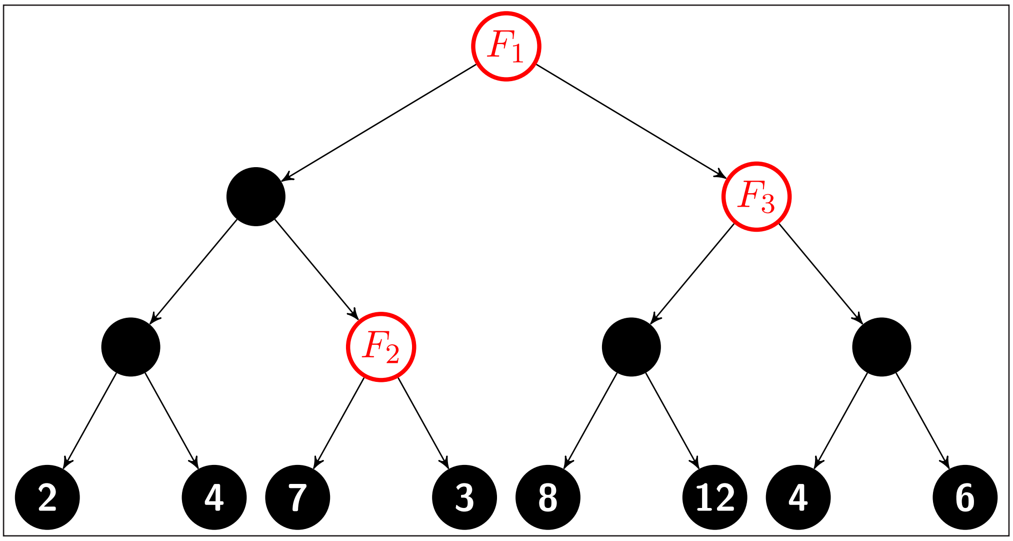


Figure 1 Each occurrence of feature F splits the dataset into two parts. In the example, F_1 creates partition classes L_1 = {2, 4, 7, 3} and R_1 = {8, 12, 4, 6}. The split at F_2 creates classes L_2 = {7} and R_2 = {3}, whereas the split at F_3 defines L_3 = {8, 12} and R_3 = {4, 6}. The model is agnostic to any features other than F.

Trend Estimator. For the empirical analysis of the datasets, we wrote a *trend estimator module*, which trains both a random forest regression model as well as a linear model on the input data. For each feature, the trend estimator then outputs

(1) the coefficient of the linear model

(2) Spearman's rank correlation and Pearson correlation coefficient between the target and

  (a) the Shapley value

  (b) the feature

(3) the absolute traversal rate (ATR)

(4) the relative traversal rate (RTR).

3.2 MEASURES OF IMPORTANCE

We compare five different notions of *feature importance*. Two definitions, the impurity-based feature importance and the permutation-based feature importance, are well studied concepts (Breiman 2001). We use *scikit-learn's* default implementation of these measures. Moreover, the exclusion-based feature importance is a folklore variant to measure a feature's importance by the model's decrease in accuracy by excluding a feature. To compare different notions of feature importance with the exclusion-based variant, we denote by $MSE(R, F)$ the mean squared error of the random forest model in which feature F was excluded. We then calculate $MSE(R, F)/MSE(R)$ as a measure of feature importance. We train 50 random forest models $R_1, \ldots, R_{50}$ independently on 80% of the data each and estimate $\hat{E}(F)$ as the mean over $MSE(R_i,F)/MSE(R_i)$ *(for* $i = 1 \ldots 50$). Finally, we define the exclusion-based feature importance of features $(F_1, \ldots, F_d)$ as the normalised value of $\hat{E}(F)$, thus $E(F) = \hat{E}(F)/(\hat{E}(F_1) + \ldots + \hat{E}(F_d))$.

In addition, we introduce two novel types of feature importance based on residual learning. The idea is that the importance of feature F_i is determined by its residuals given features $F_1, \ldots, F_{i-1}, F_{i+1}, \ldots F_d$. With a slight misuse of notation, we interpret $F_i \in \mathbb{R}^n$ as the vector of all values corresponding to feature F_i and denote by $Y \in \mathbb{R}^n$ the values of the dependent variable. We denote by $\mathcal{A}_j$ an arbitrary algorithm that takes $F_1, \ldots, F_{j-1}$ as input and outputs a vector in $\mathbb{R}^n$. Given a fixed permutation π of $[d]$, we denote by $i_1, \ldots, i_d$ the new order under π. To determine the importance of F_i, we determine its importance under all permutations π with the property that $i_d = i$ and weight it by the performance of a model consisting only of the feature F_i. The interpretation is as follows: given all other features, what can be learned from feature F_{i_d}*?* The algorithm to compute the importance can now be expressed as follows.

- For all permutations π which map $i \longmapsto d$, do the following
  - Define $W_1^\pi = F_{i_1}$.
  - Replace the values of feature F_{i_j} with $W_{i_j}^\pi = F_{i_j} - \mathcal{A}_{\rangle|}(W_{i_1}^\pi, \ldots, W_{i_{j-1}}^\pi)$ (for $j = 2 \ldots d$).
  - Train a random forest with features $\{W_{i_j}\}$.
  - Determine the impurity-based feature importance of $W_{i_d}^\pi$.
- Determine the average feature importance of F_i as the mean over all $W_{i_d}^\pi$, call this $(FI)_i$.
- Train a random forest regressor $\mathcal{R}_i$ with feature F_i and dependent variable Y and measure $r(\mathcal{R}(F_i),Y)$.
- Return $\tilde{f}_i = r(\mathcal{R}_i(F_i),Y)(FI)_i$

After applying this algorithm, we are left with $\tilde{f}_1, \ldots, \tilde{f}_d$. Finally, we define the feature importance based on the residual algorithm $\mathcal{A}$ as the standardized version of the above estimator, namely:

$$f_j(\mathcal{A}) = \frac{\tilde{f}_j}{\Sigma_{i=1}^d \tilde{f}_i}.$$

Formally, the algorithm is given as Algorithm 1. We note the following.

- $f_j(\mathcal{A})$ is a random quantity because it depends on the training of the random forest regressors $\mathcal{R}_1, \ldots, \mathcal{R}_d$ and the random forest regressors using the features $\{W_{i_j}\}$.
- In applications, it may not be possible to iterate over all permutations π. Instead, the average impurity-based feature importance is estimated by sampling some permutations.
- The algorithm is highly dependent on the residual algorithm $\mathcal{A}$.

In this contribution, we empirically analyze the feature importance based on two different residual algorithms: classical residual learning by random forest regression and decorrelation by the Gram-Schmidt method.

```
Require: $d$ features $F_1, \dots, F_d$, residual algorithm, dependent variable $Y$
    $S_d \leftarrow$ set of permutations of $\{1, 2, \dots, d\}$
    FeatImp $\leftarrow (0, \dots, 0) \in \mathbb{R}^d$
    for $\pi \in S_d$ do
        $F_{i_j} \in \mathbb{R}^n \leftarrow j$-th feature vector under permutation $\pi$
        $W^{\pi}_{i_1} = F_{i_1}$
        for $j = 2 \dots d$ do
            $W^{\pi}_{i_j} \leftarrow F_{i_j} - \mathcal{A}_{i_j}(W^{\pi}_{i_1},\dots,W^{\pi}_{i_{j-1}})$
        end for
        RF $\leftarrow$ generate a random forest model with features $W^{\pi}_{i_1},\dots,W^{\pi}_{i_d}$ and dependent variable $Y$
        FI $\leftarrow$ result of impurity-based feature importance of RF for feature $W^{\pi}_{i_d}$
        $\mathcal{R}_{i_d} \leftarrow$ generate random forest model with feature $F_{i_d}$ and dependent variable $Y$
        $\mathrm{FeatImp}[k] \leftarrow \mathrm{FeatImp}[k] + r(\mathcal{R}_{i_d}(F_{i_d}), Y) \cdot \mathrm{FI}$
    end for
    $\mathrm{FeatImp} \leftarrow \frac{\mathrm{FeatImp}}{\|\mathrm{FeatImp}\|_1}$
    return FeatImp
```

Algorithm 1 Residual-based feature importance.

Residual learning-based feature importance

Following Dezfouli et al. (2019), it is a natural idea to define the family of residual algorithms $\mathcal{A}_2,\dots,\mathcal{A}_d$ as a family of random forest regressors. More precisely, given $W^{\pi}_{i_1},\dots,W^{\pi}_{i_{j-1}}$, we train a random forest regressor $\mathcal{R}$ on those features with the dependent variable F_{i_j}. Hence,

$$\mathcal{A}_j(W^{\pi}_1,\dots,W^{\pi}_{j-1}) = \mathcal{R}(W^{\pi}_1,\dots,W^{\pi}_{j-1}).$$

Thus, we subtract from F_{i_j} everything that can be learned by random forest regressors from the first j–1 features under π. This approach is classically known as residual learning and finds prominent applications in machine learning (He et al. 2016).

Gram-Schmidt decorrelation-based feature importance

Another natural approach is to use the very famous Gram-Schmidt orthogonalization technique. While it is a standard tool in mathematics to generate orthogonal bases of vector spaces, it was first applied in the early 2000s to find independent components in complex datasets (Zhang & Chan 2006). The most important observation is that the covariance is an inner product, so the very general Gram-Schmidt orthogonalization technique can be applied with the covariance to create decorrelated features. Here, we define:

$$\mathcal{A}_j(W^{\pi}_1,\dots,W^{\pi}_{j-1}) = \sum_{i=1}^{j-1} \frac{\mathrm{Cov}(F_j, W^{\pi}_i)}{\mathrm{Cov}(F_j, F_j)} F_j.$$

A major advantage may be that this orthogonalization method, unlike the above approach, is fully mathematically understandable. However, it may be brittle to nonlinear dependencies.

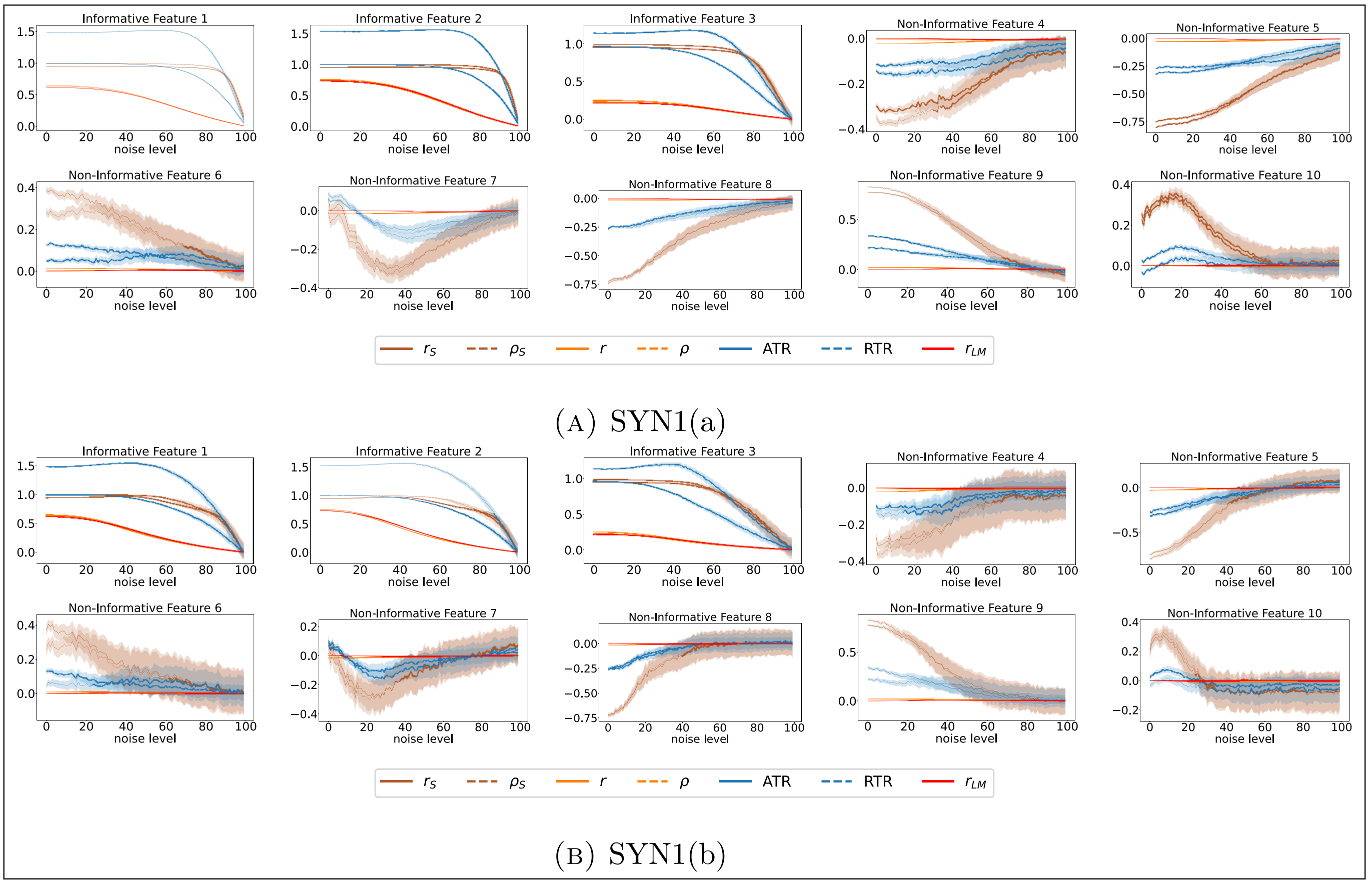
4. RESULTS

4.1 FINDING TRENDS

In the following, we report our empirical results on the performance of the different trend estimators on the HOUSING, FISH, and SYN1 datasets.

4.1.1 SYN1

The datasets SYN1(a) and SYN1(b) were used to test the robustness of the different trend estimators with respect to the mixing of the dataset with noise. To do this, the trend estimator module was applied to D_w for each $w \in [0.01, ..., 1.]$ for each of the 250 random noise datasets. In our experiment, the aggregated output shows that both ATR and RTR, as well as the Shapley correlation, are more robust than the linear model for the informative features for both Gaussian noise and uniform white noise (Figure 2). The Shapley values are the most robust, followed by RTR and ATR. Interestingly, non-informative features were also assigned large ρ_s, r_s, RTR and ATR values.



(A) SYN1(a)

(B) SYN1(b)

Figure 2 Mean and 95% confidence interval for the different trend estimators on SYN1(a) and SYN(b) for 250 independent trials each. On the x-axis, the proportion of noise is reported. Features 1–3 are informative, whereas features 4–10 are non-informative.

4.1.2 FISH

We performed three experiments on the FISH dataset using the features *weight*, *height* and *width* to predict *length*. All three selected features are positively correlated with the target (Figure 3).

First, we applied the trend estimation module to the FISH dataset. To control for random effects, we performed 100 bootstrapping iterations, sampling from a subset of 70%. The linear regression model assigned a negative coefficient to the *Height* feature, while the other trend estimators reported a positive trend (Figure 4).

To evaluate the robustness of the trend estimators to noise, we used a random mixing strategy similar to that used to create SYN1. The FISH data were standardized and mixed with random noise ranging from 0% to 99% noise before being used as input to the trend estimator module. We found that the linear model and the RTR became unstable as the feature-to-target correlation r and ρ decreased, while the ATR and the Shapley measures r_s and ρ_s remained relatively unaffected up to much higher mixing rates (Figure 5).

4.1.3 HOUSING

The random forests were trained 100 times on HOUSING, each run using a random initialization and a random subset containing 70% of the training data. The mean and standard deviation of the different trend estimators, as well as the relative absolute SHAP values is shown in Figure 6, i.e. the sum of the absolute SHAP values divided by the maximum absolute sum for

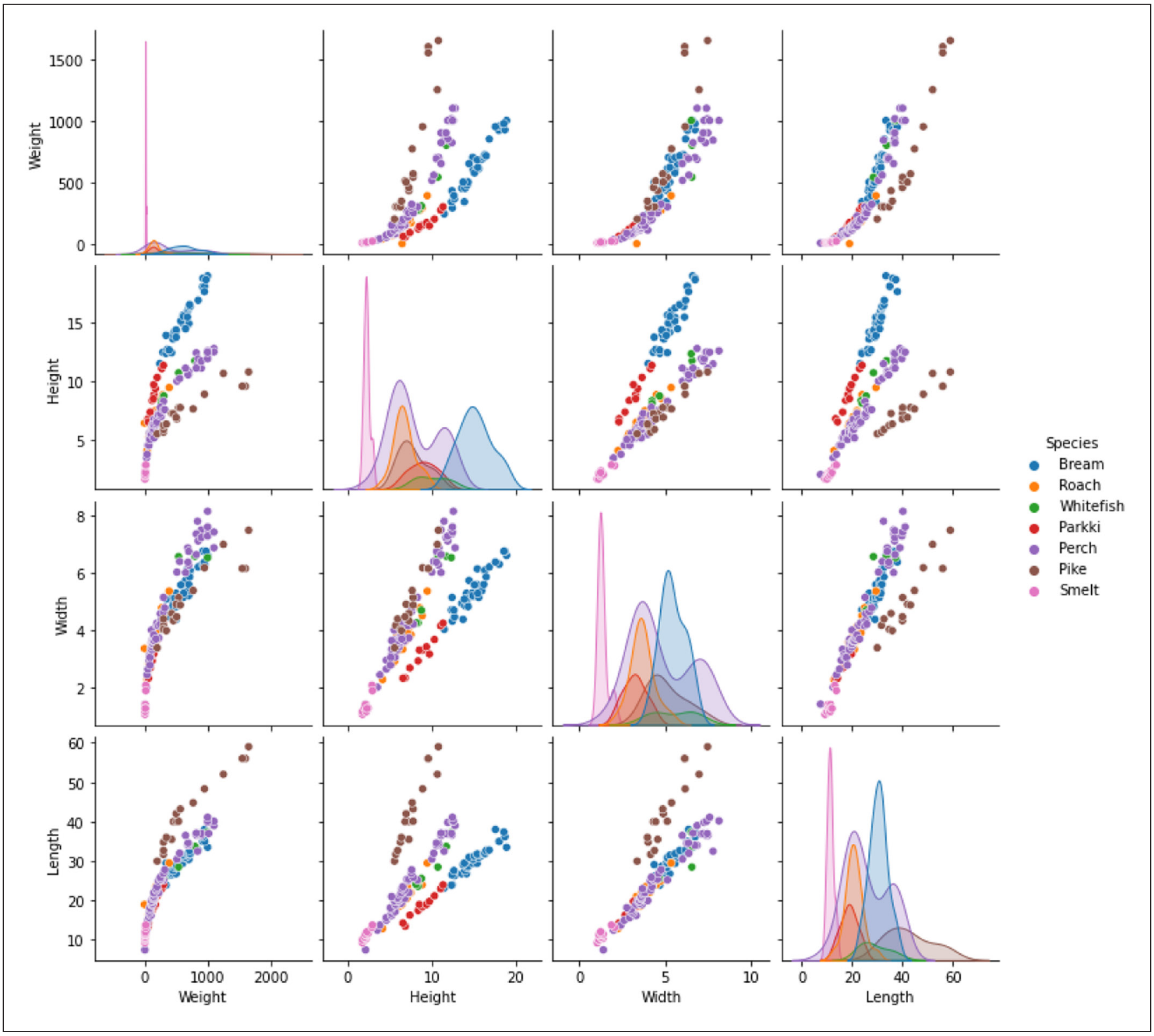


Figure 3 Pairplot of the used fish market dataset features (*weight, height and width*) and the predicted variable (Length).

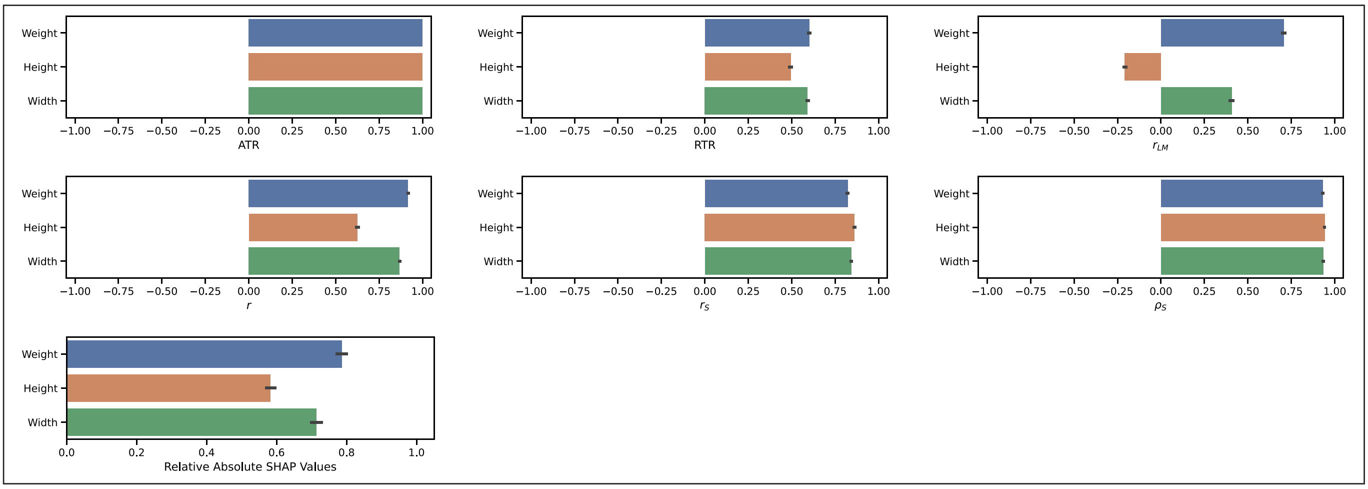


Figure 4 Comparison of the trend estimators for FISH. We report the mean and the standard deviation of the different trend estimators over 100 bootstrap iterations, each containing 70% of the data. Relative absolute SHAP values shows the absolute sum of the SHAP values for each run, divided by the highest respective sum.

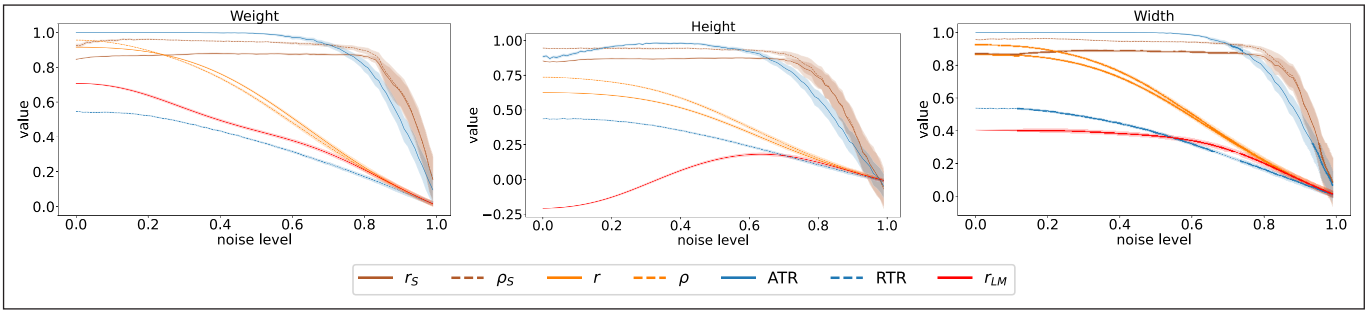


Figure 5 Mean and 95% confidence interval w.r.t. 100 independent iterations over noise on FISH. The x-axis reports the proportion of noise mixed to the real data.

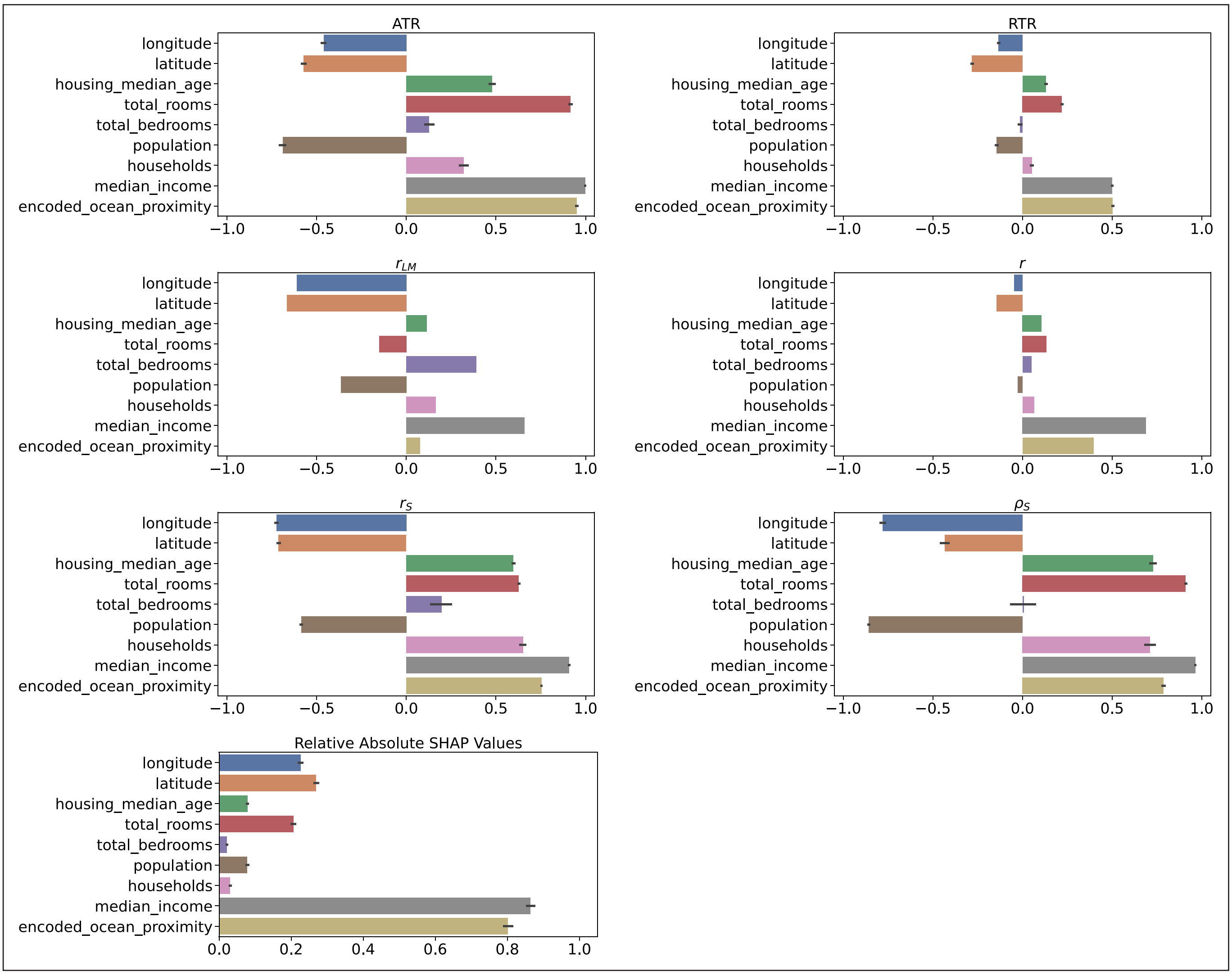


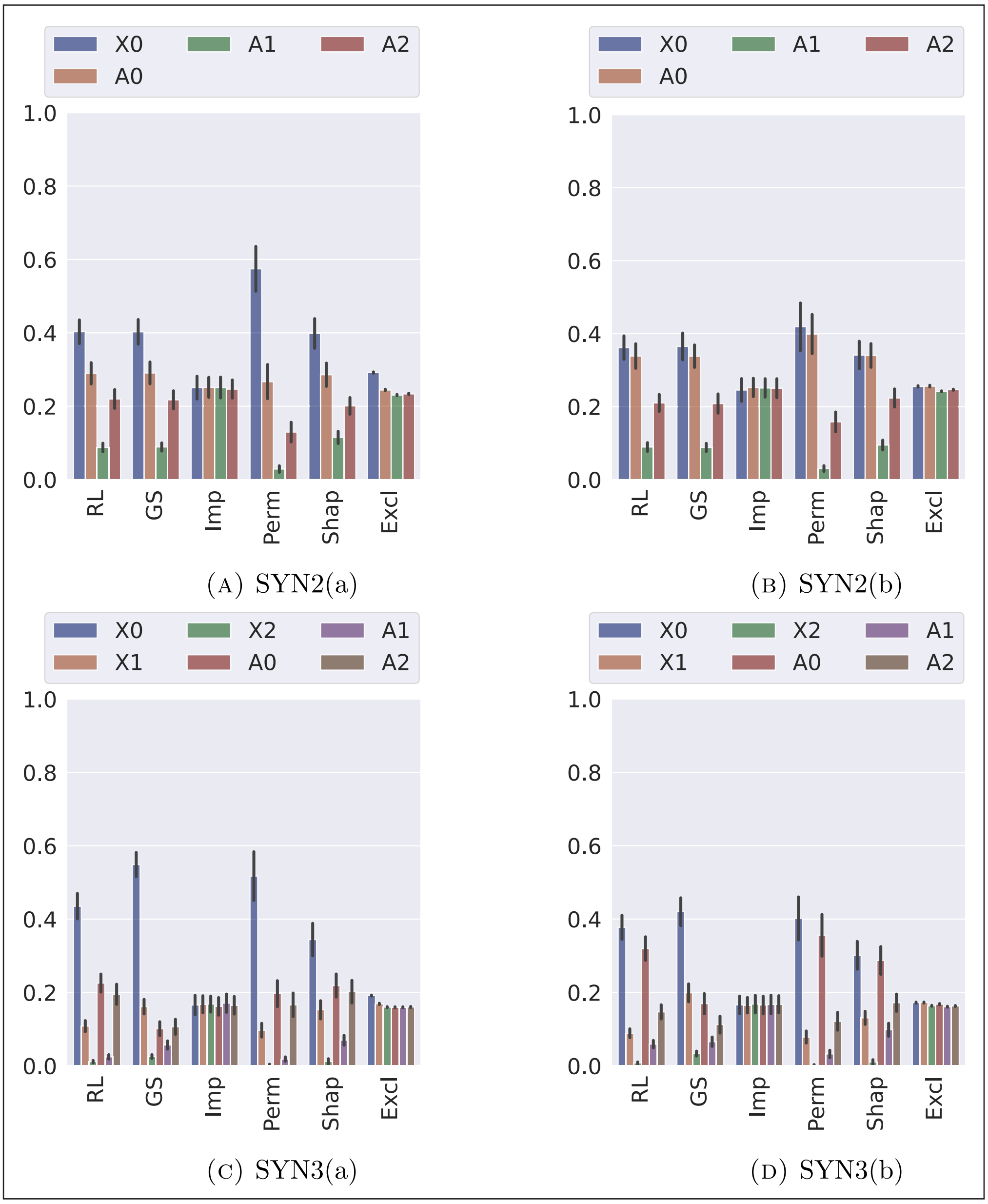
Figure 6 Comparison of the trend estimators on HOUSING. The linear model assigns a negative coefficient to the total number of rooms feature, even though the feature itself is positively correlated to the target.

each run. The characteristic *population* is very weakly negatively correlated with the housing price. However, all trend estimators report a significant negative trend for population. The feature total rooms is positively correlated with the target. However, the linear model assigns a negative coefficient to the total number of rooms. All other trend estimators report a positive trend.

4.2 MEASURES OF IMPORTANCE

We compare the impurity-based feature importance, the permutation-based feature importance, the exclusion-based feature importance and the feature importance induced by the two described residual algorithms (residual learning and Gram-Schmidt decorrelation). A run consists of fitting a random forest, to measure the feature importance values, multiple independent runs (between 100 and 400) were conducted. To determine the residual-based importance scores, for each feature, 50 permutations that assign this feature to the last position are sampled independently in every run.

First, we compare the different scores on the synthetic datasets SYN2 and SYN3 (see Figure 7). Perhaps the most important observation is that the impurity-based feature importance assigns the same score to all features – in both datasets. This is in strong contrast to all other feature importance scores. It is noteworthy that both residual-based approaches produce very comparable scores on the given datasets. Both residual-based approaches and the permutation-based score assign roughly the same score to X_0 and the slightly noisy variant A_0. However, feature A_1, which is subject to much more noise, receives a significantly higher score under residual-based scoring. Especially on SYN3, the residual-based approaches assign a not too small score to all informative features. The permutation-based score for the informative features X_1 and X_2 is comparatively small. However, all scores assign a higher importance to the noisy instance A_0 of X_0 than to the informative features X_1 and X_2.



(A) SYN2(a)

(B) SYN2(b)

(C) SYN3(a)

(D) SYN3(b)

Figure 7 Comparison of the six different notions of feature importance on synthetic data. Figures A and B show results with respect to SYN2(a) and SYN2(b). Here, the labels are generated as $Y = 4 \cdot X_0^{1.5}$, and $\{A_i\}$ are given as by $X_0 + \mathcal{W}_i$ for differently strong Gaussian noise $\mathcal{W}_i$ (SYN2(a)) and white noise (SYN2(b)). Figures C and D show results with respect to SYN3(a) (Gaussian noise) and SYN3(b) (White noise). Here, the labels are generated as $Y = 4 \cdot X_0^{1.5} + 2 \cdot X_1 + 0.5 \cdot X_2^2$, thus two more (weakly) informative features are given.

Next, we compare the different scores on the real datasets HOUSING and FISH (see Figure 8). For HOUSING, it is most striking that the residual learning, impurity-based, and permutation-based scores assign the largest value to the median income, followed by the proximity to the ocean and the latitude/longitude, while the Gram-Schmidt-based score assigns only a large value to the median income and all other features receive comparable scores. In addition, the population is found to be more important by the impurity-based and permutation-based approaches as opposed to the residual learning-based approach.

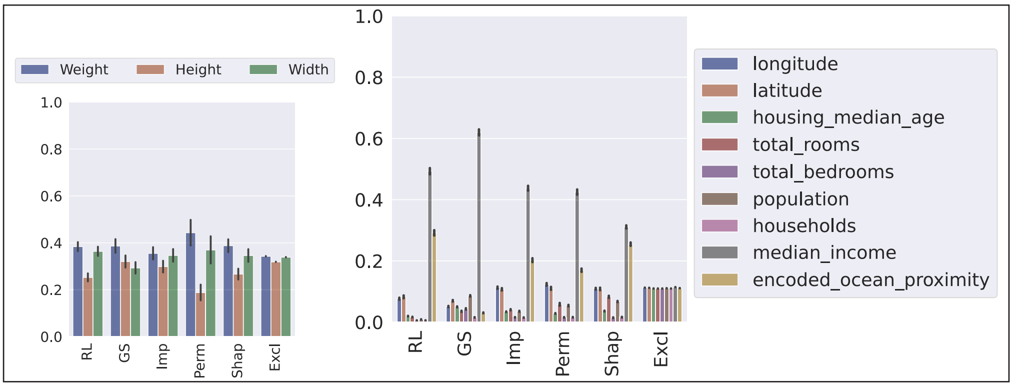


Figure 8 Comparison of the six different notions of feature importance on real-world instances. The l.h.s. reports the feature importance scores on the FISH dataset (mean and standard deviation over 400 independent runs), the r.h.s. on HOUSING (mean and standard deviation over 100 independent runs).

For FISH, all measures assign the highest score to weight and all measures assign a non-vanishing score to all three variables. However, the score of width is within one standard deviation in the residual learning-based, impurity-based, and permutation-based approaches. Only the Gram-Schmidt-based score assigns a significantly larger value to weight and considers height to be the second most important feature.

5. CONCLUSION

We present two novel estimators for monotone trends in a dataset based on random forest regression. They perform much more reliably than the often proposed linear model coefficient and are robust to noise. However, the SHAP values perform equally well and are much better understood from a theoretical point of view. Nevertheless, we believe that the traversal rate-based approach has its merits. It depends only on the random forest model (trained on some dataset) and the computation is completely independent of the specific data, once the model exists. This means, in particular, that given the random forest model (which has to be trained, and fine-tuned reasonably well), the trends estimated by the traversal rates do not vary in terms of input data. SHAP values, on the other hand, are computed as a combination of the model and some data (which may also have its own merits). Particularly, SHAP plots look different if different data points are included to calculate the SHAP values. The calculation of a Shapley-value is done by calculating and comparing different coalitions of features, where the traversal rate approach is agnostic with regard to other features. If only an adversarial part of the full dataset is used to generate SHAP plots, they can look completely different and assign different trends to the features. On the positive side, the SHAP analysis also provides insights into new data points during prediction: we can easily track which feature's values increase and, respectively, decrease the model's prediction. With respect to feature importance, we introduced the residual-based approach. We compared the results on synthetic data and two real instances. It is noteworthy that both residual-based approaches produce comparable results on the synthetic datasets, but this may be due to the fact that the noise is added linearly. Overall, the residual-based approaches perform much better on highly correlated features than the impurity-based approach. Their results are comparable to the permutation-based approach in many facets. However, significant differences were also found. In particular, informative features that contribute weakly to the noise were assigned higher values than by the permutation-based score. Therefore, we believe that the residual-based feature importance scores should be preferred for use on datasets with highly dependent features.

FUNDING INFORMATION

All authors were supported by the German Research Council, grant DFG FOR 2975. We gained support from OA Open Access publication fund of Goethe University Frankfurt.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Yannick Gerstorfer
Frankfurt Institute for Advanced Studies, Goethe University Frankfurt Frankfurt 60325, Germany

Max Hahn-Klimroth
Faculty of Computer Sciences, TU Dortmund University Dortmund 44227, Germany

Lena Krieg
Faculty of Computer Sciences, TU Dortmund University Dortmund 44227, Germany

REFERENCES

Beraha, M, Metelli, AM, Papini, M, Tirinzoni, A and **Restelli, M.** 2019. Feature selection via mutual information: New theoretical insights. In: *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, pp. 1–9. DOI: https://doi.org/10.1109/IJCNN.2019.8852410

Boulesteix, AL, Janitza, S, Kruppa, J and **König, IR.** 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6): 493–507. DOI: https://doi.org/10.1002/widm.1072

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32. DOI: https://doi.org/10.1023/A:1010933404324

Chen, RC, Dewi, C, Huang, SW and **Caraka, RE.** 2020. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1): 52. DOI: https://doi.org/10.1186/s40537-020-00327-4

Chicco, D and **Jurman, G.** 2021. An ensemble learning approach for enhanced classification of patients with hepatitis and cirrhosis. *IEEE Access*, 9: 24485–24498. DOI: https://doi.org/10.1109/ACCESS.2021.3057196

Dezfouli, A, Ashtiani, H, Ghattas, O, Nock, R, Dayan, P and **Ong, CS.** 2019. Disentangled behavioural representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 32. DOI: https://doi.org/10.1101/658252

Fraser, DAS. 1965. On information in statistics. *The Annals of Mathematical Statistics*, 36(3): 890–896. DOI: https://doi.org/10.1214/aoms/1177700061

Gübert, J, Hahn-Klimroth, M and **Dierkes, PW.** 2023. A large-scale study on the nocturnal behavior of African ungulates in zoos and its influencing factors. *Frontiers in Ethology*, 2: 1219977. DOI: https://doi.org/10.3389/fetho.2023.1219977

Guyon, I, Weston, J, Barnhill, S and **Vapnik, V.** 2002. Gene selection for cancer classification using Support Vector Machines. *Machine Learning*, 46(1–3): 389–422. DOI: https://doi.org/10.1023/A:1012487302797

He, K, Zhang, X, Ren, S and **Sun, J.** 2016. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

Janzing, D, Minorics, L and **Bloebaum, P.** 2020. Feature relevance quantification in explainable ai: A causal problem. Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. In: *Proceedings of Machine Learning Research (PMLR)*, 108: 2907–2916. Available at https://proceedings.mlr.press/v108/janzing20a.html

Louppe, G. 2014. Understanding random forests: From theory to practice. arXiv:1407.7502.

Lundberg, SM and **Lee, SI.** 2017. A unified approach to interpreting model predictions. In: *31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4768–4777.

Nugent, C. 2017. California housing prices. version 1. Available at https://www.kaggle.com/datasets/camnugent/california-housing-prices [accessed 24 February 2023].

Parveen, AN, Inbarani, HH and **Kumar, ENS.** 2012. Performance analysis of unsupervised feature selection methods. In: *2012 International Conference on Computing, Communication and Applications*, Dindigul, India, pp. 1–7. DOI: https://doi.org/10.1109/ICCCA.2012.6179181

Pyae, A. 2019. Fish market dataset. version 2. Available at: https://www.kaggle.com/datasets/aungpyaeap/fish-market [accessed: 24 February 2023].

Saarela, M and **Jauhiainen, S.** 2021. Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, 3(2). DOI: https://doi.org/10.1007/s42452-021-04148-9

Shapley, LS. 1953. 17. A Value for n-Person Games. In: *Contributions to the Theory of Games (AM-28)*, Volume II. Princeton: Princeton University Press, pp. 307–318. DOI: https://doi.org/10.1515/9781400881970-018

Sundararajan, M and **Najmi, A.** 2020. The many shapley values for model explanation. Proceedings of the 37th International Conference on Machine Learning. In: *Proceedings of Machine Learning Research,* 119: 9269–9278. Available at https://proceedings.mlr.press/v119/sundararajan20b.html.

Zhang, K and **Chan, LW.** 2006. Dimension reduction as a deflation method in ICA. *IEEE Signal Processing Letters*, 13(1): 45–48. DOI: https://doi.org/10.1109/LSP.2005.860541

TO CITE THIS ARTICLE:
Gerstorfer, Y, Hahn-Klimroth, M and Krieg, L. 2023. A Notion of Feature Importance by Decorrelation and Detection of Trends by Random Forest Regression. *Data Science Journal*, 22: 42, pp. 1–14. DOI: https://doi.org/10.5334/dsj-2023-042

Submitted: 26 May 2023
Accepted: 27 September 2023
Published: 03 November 2023

Data Science Journal is a peer-reviewed open access journal published by Ubiquity Press.