applied sciences

Article

Feature Interaction in Terms of Prediction Performance

Sejong Oh

Department of Software Science, Dankook University, Yongin 16894, Korea; sejongoh@dankook.ac.kr

Received: 24 September 2019; Accepted: 26 November 2019; Published: 29 November 2019

Abstract: There has been considerable development in machine learning in recent years with some remarkable successes. Although there are many high-performance methods, the interpretation of learning models remains challenging. Understanding the underlying theory behind the specific prediction of various models is difficult. Various studies have attempted to explain the working principle behind learning models using techniques like feature importance, partial dependency, feature interaction, and the Shapley value. This study introduces a new feature interaction measure. While recent studies have measured feature interaction using partial dependency, this study redefines feature interaction in terms of prediction performance. The proposed measure is easy to interpret, faster than partial dependency-based measures, and useful to explain feature interaction, which affects prediction performance in both regression and classification models.

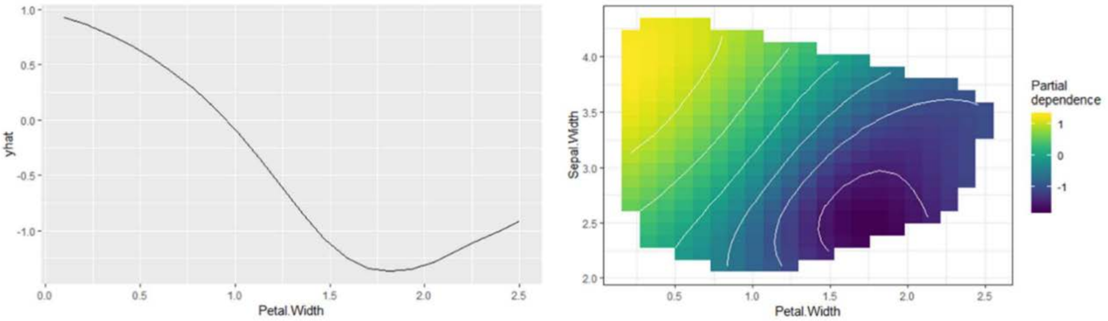
Keywords: machine learning; black box model; model interpretability; feature interaction; feature importance

1. Introduction

The emerging technology report published by Gartner in 2017 identifies that machine learning is currently located at the peak of inflated expectations [1]. Machine learning techniques have been adopted by an increasing number of people, which has led to success stories in many fields. Schwartz states that machine learning is no longer restricted to the experts in the Harvard Business Review [2]. Although machine learning has seen considerable success in recent years, most experts in this field still consider the prediction models generated by machine learning algorithms to be "black boxes". They do not have knowledge about how specific prediction results are obtained and the underlying principle behind the functioning of machine learning models. The concepts behind regression and decision tree models are relatively easy to understand, whereas the interpretation of those based on neural networks and support vector machines are more complex. Thus, researchers have concentrated their efforts on explaining the working mechanism of learning models and their prediction results [3].

Feature importance, the partial dependence plot (PDP) [4], feature interaction, and the Shapley value [5] have been proposed to achieve model interpretation. All of the above stated features can be calculated and interpreted only after model fitting. Feature impact or feature importance [6–8] measures the extent to which the feature or variable influences the prediction results of the given model. A prediction model is built on multiple features, wherein each feature makes a unique contribution toward the prediction. A model can be better understood by having knowledge about the influence of each feature in the model. Various methods have been proposed to measure feature importance. Recently permutation-based feature importance, which measures the importance of a feature *f1* by permuting its feature values, was introduced by Breiman [9] and Fisher et al. [10]. This disconnects *f1* from the other features and changes the predictive performance of the given model. The importance/significance of *f1* increases with a greater degree of change in the prediction performance.

The partial dependence plot, which was introduced by Friedman, shows the marginal effect of one or two features on the predictive outcome of a machine learning model [4]. The PDP helps in determining the transition/change in the predictive performance in relation to the change in the feature data values (observations). Furthermore, it is possible to obtain the range of data values in a feature that is either useful for prediction or shows low prediction performance (see Figure 1).



(a) PDP of a single feature

(b) PDP of two features

Figure 1. Example of a partial dependence plot (PDP). (**a**) The data values in the range [0, 1] of a feature "Petal.Width" show high prediction performance, while those in the range [1.5, 2.0] show low prediction performance. (**b**) The light yellow region indicates the data range of two features that produces a high prediction performance, whereas the dark blue region represents a low prediction performance data range.

Features in a prediction model tend to collaborate in a prediction task, which indicates the presence of feature interaction. Various methods for measuring feature interaction have been proposed. Initially, statistics-based methods were presented. Hastie and Tibshirani [11] suggested measuring feature interaction based on the analysis of variance (ANOVA) test. After the ANOVA test, the corresponding *p*-value for each pair of features is computed to measure feature interaction. This approach requires extensive computation time. Loh and Lou et al. [12,13] tested pairwise feature interactions based on the χ^2 test. Sorokina et al. proposed a grove-based method to detect statistical interactions [11]. Decision-tree-based models, such as "random forest", measure feature interaction using a tree structure. If features f_1 and f_2 are located on the same path of a decision tree, they can be regarded as interacting [14–16].

Recently, partial dependency-based methods have been proposed. The H-statistics method by Friedman and Popescu [17] is a typical method. The H-statistics used to measure the interaction between feature A and B are defined by

$$H^2_{AB} = \sum_{i=1}^{n} \left[PD_{AB}(x_A^{(i)}, x_B^{(i)}) - PD_A(x_A^{(i)}) - PD_B(x_B^{(i)})\right]^2 / \sum_{i=1}^{n} PD^2_{AB}(x_A^{(i)}, x_B^{(i)}) \quad (1)$$

where $PD()$ is a partial dependence function.

The variable interaction network method, introduced by Hooker, decomposes the prediction function into the main effects and feature interactions [7]. Greenwell et al. introduced the partial dependence-based feature importance and interaction measures [8], which facilitates the measurement between two features. This measures feature interaction between feature A and B with the following expressions:

$$\mathrm{S}_A = sd(\mathrm{i}(A)|B_\mathrm{j}) \quad (2)$$

$$\mathrm{S}_B = sd(\mathrm{i}(B)|A_\mathrm{j}) \quad (3)$$

$$\mathrm{Interact}(A,B) = (\mathrm{S}_A + \mathrm{S}_B)/2 \quad (4)$$

where sd is a standard deviation; i(A) is a feature importance of A; j = 1, 2, 3, . . . , n; and A_i is an i-th feature value of A.

The Shapley value, which was introduced by Shapley, shows the contribution of each feature toward the interpretation of a single prediction result [5]. Each feature in the Shapley value is considered to be a "player" in a game. The Shapley value ensures a fair distribution of the "payout" among all features. Figure 2 shows an example of a Shapley value plot.

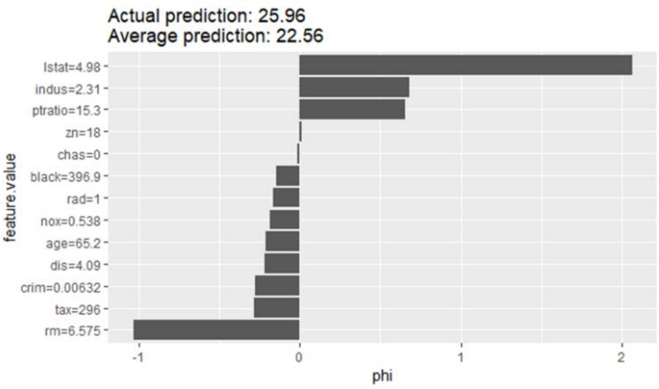


Figure 2. Example of a Shapley value plot. The feature value makes a significant contribution to the prediction if it has a phi >0, whereas a negative phi value reduces its contribution.

Molnar has presented an in-depth discussion of model interpretability that covers various methods discussed above [18] as well as implementing the iml package [19] for R and Python that contains various functions related to model interpretability.

In this paper, a new measure for feature interaction based on prediction performance is proposed. This study tries to understand feature interaction as a variation of prediction performance. In recent studies, feature interaction has been measured using partial dependency, wherein feature combination is used to measure the partial dependency variance [7,8,17]. Although the concepts of partial dependency are clearly known, it is difficult to intuitively understand how it impacts prediction performance. This necessitates the need for more lucid feature interaction measures to explain prediction performance. Prediction performance has emerged as a new candidate in feature interaction measurement. While various combinations of feature interactions exist in a model, this study will focus on the interaction between any two features.

2. Proposed Interaction Measure

The terms "prediction performance" and "feature interaction" are discussed below prior to introducing the proposed method. In this study, two types of prediction tasks, namely, regression and classification, were considered. In the regression task, the target value of a prediction is a continuous value. In the classification task, the target value of a prediction is a categorical value. The classification accuracy used to measure classification performance is defined as follows:

$$Acc = \frac{\text{Number of correctly predicted cases}}{\text{Total number of predicted cases}}. \tag{5}$$

The root mean square error (RMSE) used to measure regression performance is defined as follows:

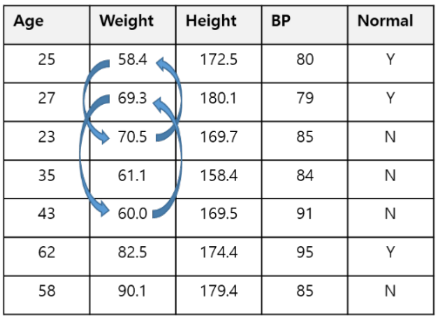
$$RMSE = \sqrt{\left(\sum\nolimits_{i=1}^{n} (Actual\ value_i - predicted\ value_i)^2\right)/n}. \tag{6}$$

Thus, in the context of this study, the prediction performance refers to the classification accuracy or RMSE with respect to the type of prediction task. However, classification accuracy or RMSE is not the only performance measure. The proposed method is highly compatible with other measures such as the area under the curve (AUC) value.

Feature interaction can be described as a phenomenon that involves two or more features meeting and influencing each other during a prediction task. Owing to the presence of feature interaction, the overall prediction performance of a model is not equal to a simple sum of the performance of each constituent feature. Feature interaction can be measured by determining the variance in the prediction performance. The following notations will be defined prior to introducing the formal definition of feature interaction:

Notations
M: the fitted prediction model (regression or classification)
DS: the given training dataset used to build a prediction model M
F_U: all features of a given dataset DS. $F_U = \{F_1, F_2, F_3, \ldots, F_n\}$
$PP_M(X)$: prediction performance of a given dataset X under the fitted model M
$DS\_Perm(F_i)$: dataset consisting of a permuted feature $F_i \in F_U$

Feature permutation involves a random shuffle of the feature values. Figure 3 illustrates a permuting feature denoted by "weight". Feature permutation neutralizes the influence of a given feature in a prediction model.



Age	Weight	Height	BP	Normal
25	58.4	172.5	80	Y
27	69.3	180.1	79	Y
23	70.5	169.7	85	N
35	61.1	158.4	84	N
43	60.0	169.5	91	N
62	82.5	174.4	95	Y
58	90.1	179.4	85	N

Figure 3. An example of feature permutation.

Definition 1. *The reduced performance error Err obtained by removing feature F_i is defined as*

$$Err(F_i) = \begin{cases} PP_M(DS) - PP_M(DS\_Perm(F_i)), \text{if } M \text{ is a classification model} \\ PP_M(DS\_Perm(F_i)) - PP_M(DS), \text{if } M \text{ is a regression model} \end{cases} \quad (7)$$

In a classification task, PP_M represents the classification accuracy for the model M. The use of a dataset with a permuted feature F_i ($DS\_Perm(F_i)$) as opposed to the original dataset DS may result in reduced accuracy. Thus, the reduced accuracy can be interpreted as an "error" caused by the permuted feature F_i. A large value of $Err(F_i)$ indicates the importance of F_i in the given prediction task. A negative $Err(F_i)$ value suggests that feature F_i hinders/affects accurate prediction. In such cases, F_i can be removed from the training set DS to improve the performance of the model. In a regression task, PP_M represents the RMSE value for the model M. The dataset consisting of a permuted feature F_i ($DS\_Perm(F_i)$) may result in a higher RMSE compared to the original dataset DS. This indicates that the equation describing $Err(F_i)$ is different for the regression and classification tasks.

Previous studies have used the concept of "permuted feature" to measure the feature impact [6–8]. In this study, the "permuted feature" is used as a measure of performance reduction, implying that both feature impact and performance reduction refer to the same concept.

Definition 2. *The reduced performance Err obtained by removing the feature set* $\{F_i, F_j\} \subset F_U$ *is defined as*

$$Err\big(\{F_i, F_j\}\big) = \begin{cases} PP_M(DS) - PP_M\big(DS\_Perm\big(\{F_i, F_j\}\big)\big), \text{if } M \text{ is a classification model} \\ PP_M\big(DS\_Perm\big(\{F_i, F_j\}\big)\big) - PP_M(DS), \text{if } M \text{ is a regression model} \end{cases} \tag{8}$$

The reduced performance of two features can be defined in the context of a single feature. We can permute the two features and consider the modified dataset *DS_Perm*($\{F_i, F_j\}$). Definition 2 can also be extended to include three or more features of *Err*().

Definition 3. *The feature interaction Interact between features* F_i *and* F_j *is defined as*

$$\begin{cases} \big(Err(F_i) + Err\big(F_j\big)\big) - Err\big(\{F_i, F_j\}\big), \text{ if } M \text{ is a classification model} \\ Err\big(\{F_i, F_j\}\big) - \big(Err(F_i) + Err\big(F_j\big)\big), \text{ if } M \text{ is a regression model} \end{cases} \tag{9}$$

In Definition 3, feature interaction is defined as the difference in the performance reduction. *Err* ($\{F_i, F_j\}$) is equal to $Err(F_i)$ + Err (F_j) only when there is no interaction between the features F_i and F_j. In the context of this study, the prediction performance is influenced by the feature interaction. The prediction errors caused by F_i and F_j independently decrease/increase when there is interaction between these two features. Three cases of $Interact(F_i, F_j)$ can be defined as

$Interact(F_i, F_j) = 0$: no interaction between F_i and F_j
$Interact(F_i, F_j) > 0$: positive interaction between F_i and F_j
$Interact(F_i, F_j) < 0$: negative interaction between F_i and F_j

Negative interaction implies that the connection between F_i and F_j reduces the prediction performance, whereas positive interaction leads to an increase in the prediction performance. In other words, positive or negative interactions decrease or increase the prediction error, respectively.

3. Results

The feature interaction function described in the previous section was implemented in the R programming environment (https://www.r-project.org). The CARET package [20] was used to build the prediction model. The feature permutation was repeated 10 times, and the average of the reduced performance of the *Err*() function was computed to accurately measure the feature permutation. Only the interaction between two features was measured for simplicity.

Feature interaction was demonstrated in two sample datasets. The PimaIndiansDiabetes [21] dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases and is used to predict whether or not a patient has diabetes. It has nine features and 768 observations and is used for the classification task. The features in the PimaIndiansDiabetes dataset are described in Table 1.

The BostonHousing [22] dataset has housing data for 506 census tracts of Boston from the 1970 census. It is used for predicting the price of houses (mdev in the dataset) and is essentially a regression task. The dataset has 14 features and 506 observations. Features in the BostonHousing dataset are illustrated in Table 2.

Table 1. Description of the features in the PimaIndiansDiabetes dataset.

No	Feature Name	Description
1	Pregnant	Number of times pregnant
2	Glucose	Plasma glucose concentration (glucose tolerance test)
3	Pressure	Diastolic blood pressure (mm Hg)
4	Triceps	Triceps skin fold thickness (mm)
5	Insulin	2-Hour serum insulin (mu U/mL)
6	Mass	Body mass index (weight in kg/(height in m)^2)
7	Pedigree	Diabetes pedigree function
8	Age	Age (years)
9	Diabetes	Class variable (neg, pos)

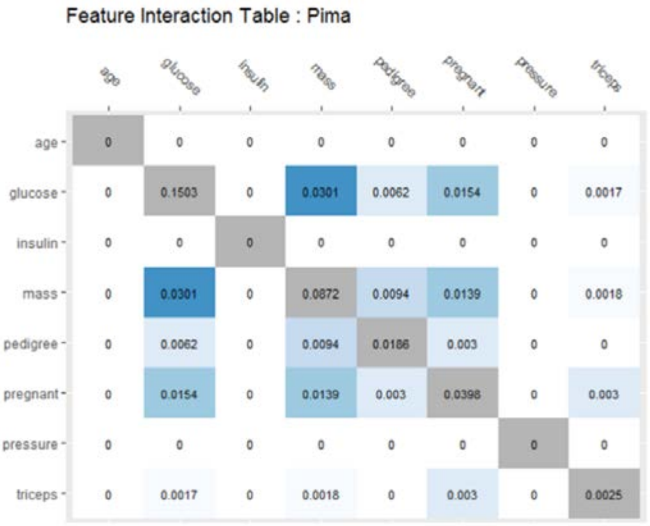
Table 2. Description of the features in the BostonHousing dataset.

No	Feature Name	Description
1	Crim	Per capita crime rate by town
2	Zn	Proportion of residential land zoned for lots over 25,000 sq.ft
3	Indus	Proportion of nonretail business acres per town
4	Chas	Charles River dummy variable (=1 if tract bounds river; 0 otherwise)
5	Nox	Nitric oxide concentration (parts per 10 million)
6	Rm	Average number of rooms per dwelling
7	Age	Proportion of owner-occupied units built prior to 1940
8	Dis	Weighted distances to five Boston employment centers
9	Rad	Index of accessibility to radial highways
10	Tax	Full-value property-tax rate per USD 10,000
11	Ptratio	Pupil–teacher ratio by town
12	B	*1000(B-0.63)^2*, where *B* is the proportion of African Americans by town
13	Lstat	Percentage of lower status of the population
14	Medv	Median value of owner-occupied homes in $1000s

The C5.0 prediction model was constructed using the PimaIndiansDiabetes dataset to test the classification, and the entire dataset was used for model fitting. The training accuracy was 0.8112. Figure 4 shows the feature interaction for various combinations of any two features in the dataset. In the interaction table, the (i-th, j-th) cell shows the feature interaction between features F_i and F_j. The reduced prediction performance due to feature F_k as calculated by the $Err(F_k)$ is shown in the cells along the (gray) diagonal (k-th, k-th). Figure 4 shows that “glucose” is the most effective feature for prediction as the *Err*(glucose) = 0.1503, which is the largest value of the *Err*() function. It means that if “glucose” is removed from the prediction model, the model performance (prediction accuracy) decreases by 0.1503. The feature “mass” offers the best interaction with “glucose” (*Interact*(glucose, mass) = 0.0301) as the prediction error decreases by 0.0301. The features “age”, “insulin”, and “pressure” have an *Err*() = 0 and *Interact*() = 0, which indicates that these features are not effective predictors of diabetes. The prediction model when rebuilt excluding the above three features resulted in an accuracy of approximately 0.80, which is approximately equal to that obtained with the original full dataset. Figure 5 shows the feature interaction between “glucose” and other features in the dataset. The feature “mass” is most interactive and “pregnant” is second. “Pressure”, “insulin”, and “age” have no interaction with “glucose”. The three features do not co-work with “glucose” to increase or decrease the prediction performance.

A linear regression model using the BostonHousing dataset was built to evaluate regression. The complete BostonHousing dataset has an RMSE of 4.6792. Figure 6 shows the interaction table where it can be observed that “lstat” (percentage of lower status of the population) is the most useful feature for predicting house price (*Err*(lstat) = 2.423). The features “b”, “crim”, “dis”, ”rad”, and “rm” are important for prediction, whereas “age” and “indus” are less influential. The regression model also displays negative interaction. Figure 7 shows that the “dis” (weighted distances to five Boston employment centers) and “rad” (index of accessibility to radial highways) features interact negatively with the “lstat”. This suggests that “dis” and “rad” decrease the prediction performance of “lstat”.

The features “rm”, “nox”, and “tax” have a positive interaction with “lstat”. They co-work with “lstat” to improve the prediction performance of the regression model.



Feature Interaction Table : Pima

	age	glucose	insulin	mass	pedigree	pregnant	pressure	triceps
age	0	0	0	0	0	0	0	0
glucose	0	0.1503	0	0.0301	0.0062	0.0154	0	0.0017
insulin	0	0	0	0	0	0	0	0
mass	0	0.0301	0	0.0872	0.0094	0.0139	0	0.0018
pedigree	0	0.0062	0	0.0094	0.0186	0.003	0	0
pregnant	0	0.0154	0	0.0139	0.003	0.0398	0	0.003
pressure	0	0	0	0	0	0	0	0
triceps	0	0.0017	0	0.0018	0	0.003	0	0.0025

Figure 4. Feature interaction table for the C5.0 model based on the PimaIndiansDiabetes dataset.

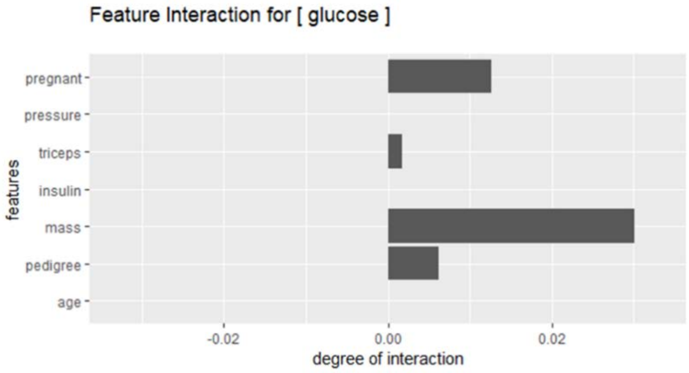
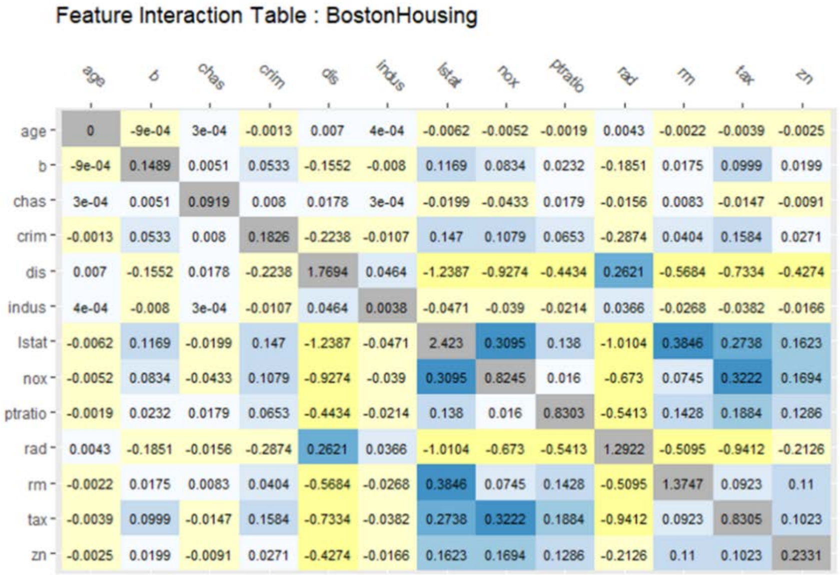


Figure 5. Feature interaction between “glucose” and other features in the PimaIndiansDiabetes dataset.



Feature Interaction Table : BostonHousing

	age	b	chas	crim	dis	indus	lstat	nox	ptratio	rad	rm	tax	zn
age	0	-9e-04	3e-04	-0.0013	0.007	4e-04	-0.0062	-0.0052	-0.0019	0.0043	-0.0022	-0.0039	-0.0025
b	-9e-04	0.1489	0.0051	0.0533	-0.1552	-0.008	0.1169	0.0834	0.0232	-0.1851	0.0175	0.0999	0.0199
chas	3e-04	0.0051	0.0919	0.008	0.0178	3e-04	-0.0199	-0.0433	0.0179	-0.0156	0.0083	-0.0147	-0.0091
crim	-0.0013	0.0533	0.008	0.1826	-0.2238	-0.0107	0.147	0.1079	0.0653	-0.2874	0.0404	0.1584	0.0271
dis	0.007	-0.1552	0.0178	-0.2238	1.7694	0.0464	-1.2387	-0.9274	-0.4434	0.2621	-0.5684	-0.7334	-0.4274
indus	4e-04	-0.008	3e-04	-0.0107	0.0464	0.0038	-0.0471	-0.039	-0.0214	0.0366	-0.0268	-0.0382	-0.0166
lstat	-0.0062	0.1169	-0.0199	0.147	-1.2387	-0.0471	2.423	0.3095	0.138	-1.0104	0.3846	0.2738	0.1623
nox	-0.0052	0.0834	-0.0433	0.1079	-0.9274	-0.039	0.3095	0.8245	0.016	-0.673	0.0745	0.3222	0.1694
ptratio	-0.0019	0.0232	0.0179	0.0653	-0.4434	-0.0214	0.138	0.016	0.8303	-0.5413	0.1428	0.1884	0.1286
rad	0.0043	-0.1851	-0.0156	-0.2874	0.2621	0.0366	-1.0104	-0.673	-0.5413	1.2922	-0.5095	-0.9412	-0.2126
rm	-0.0022	0.0175	0.0083	0.0404	-0.5684	-0.0268	0.3846	0.0745	0.1428	-0.5095	1.3747	0.0923	0.11
tax	-0.0039	0.0999	-0.0147	0.1584	-0.7334	-0.0382	0.2738	0.3222	0.1884	-0.9412	0.0923	0.8305	0.1023
zn	-0.0025	0.0199	-0.0091	0.0271	-0.4274	-0.0166	0.1623	0.1694	0.1286	-0.2126	0.11	0.1023	0.2331

Figure 6. Feature interaction table for the linear regression model based on the BostonHousing dataset.

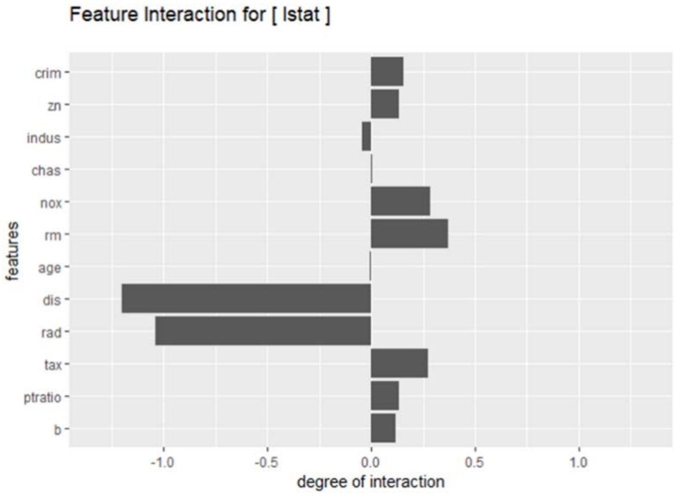
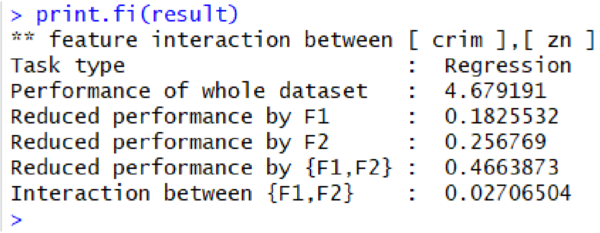


Figure 7. Feature interaction between "lstat" and other features in the BostonHousing dataset.

Apart from the main implementations in this study, namely, the two-way feature interaction plots in Figures 5 and 7 and the feature interaction tables in Figures 4 and 6, several additional functions were also implemented. The following results were obtained from the BostonHousing dataset. Figure 8 shows the feature interaction details between two features, namely, "crim" and "zn". The results show that the BostonHousing dataset is based on a regression model. The total RMSE of the complete dataset is 4.679191. Excluding the "crim"(F1) and "zn"(F2) features increases the total RMSE by 0.1825532 and 0.256769, respectively. Therefore, the prediction error will increase if "crim" and "zn" are removed from the prediction model. The total RMSE increases by 0.4663873 when both the "crim" and "zn" features are excluded. Consequently, the degree of feature interaction between the "crim" and "zn" features is equal to 0.02706504, which is calculated as 0.4663873 − (0.1825532 + 0.256769). It shows that the feature "crim" and "zn" have a positive interaction, and it means that combining "crim" and "zn" helps decrease the prediction error of house price.



```
> print.fi(result)
** feature interaction between [ crim ],[ zn ]
Task type                      :  Regression
Performance of whole dataset   :  4.679191
Reduced performance by F1      :  0.1825532
Reduced performance by F2      :  0.256769
Reduced performance by {F1,F2} :  0.4663873
Interaction between {F1,F2}    :  0.02706504
>
```

Figure 8. Feature interaction details between the "crim" and "zn" features of the BostonHousing dataset.

Figure 9 shows the total degrees of interaction corresponding to each feature. The total degree of interaction is the summation of absolute values of positive and negative interaction. The plot allows one to determine the features that strongly interact with each other. The plot also illustrates the degree of positive or negative interactions of each feature. In Figure 9, "degree_of_interaction" means the variations of prediction performance caused by each variable in a model. In the case of "dis" and "rad", they have a strong negative interaction with other features. We may consider removing the two features to decrease the prediction error of the model. The features "age", "chas", and "indus" have little interaction with other features. From Figure 6, we confirm that they have weak prediction power; thus, they are strong candidates for elimination to improve model performance.

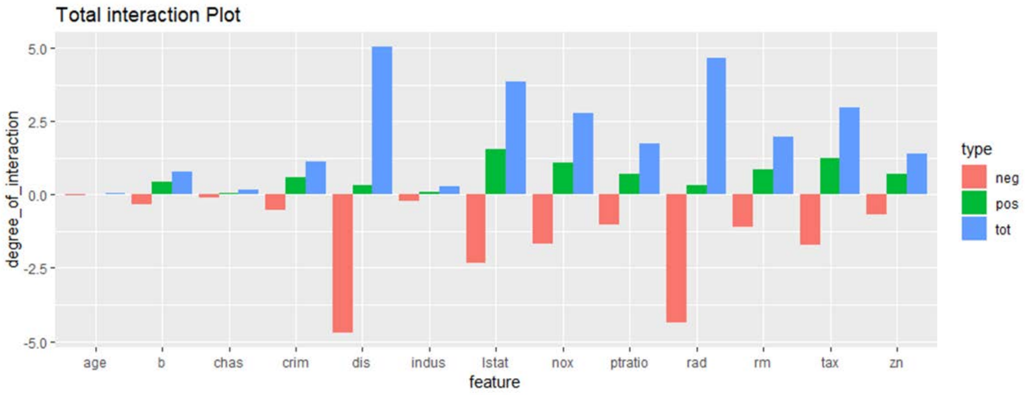
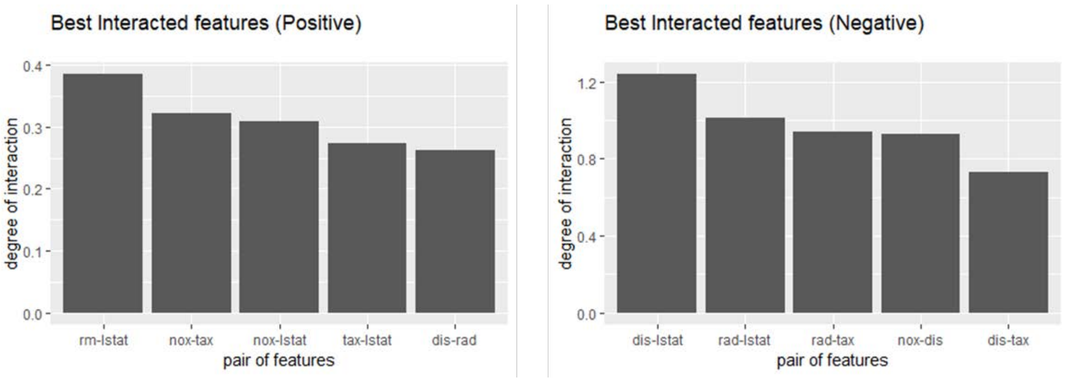


Figure 9. Total degrees of interaction corresponding to each feature.

Figure 10 shows the top five features that positively/negatively interact with each other. Positive interaction is helpful for model performance, whereas negative interaction decreases it.



(**a**) Positive interaction (**b**) Negative interaction

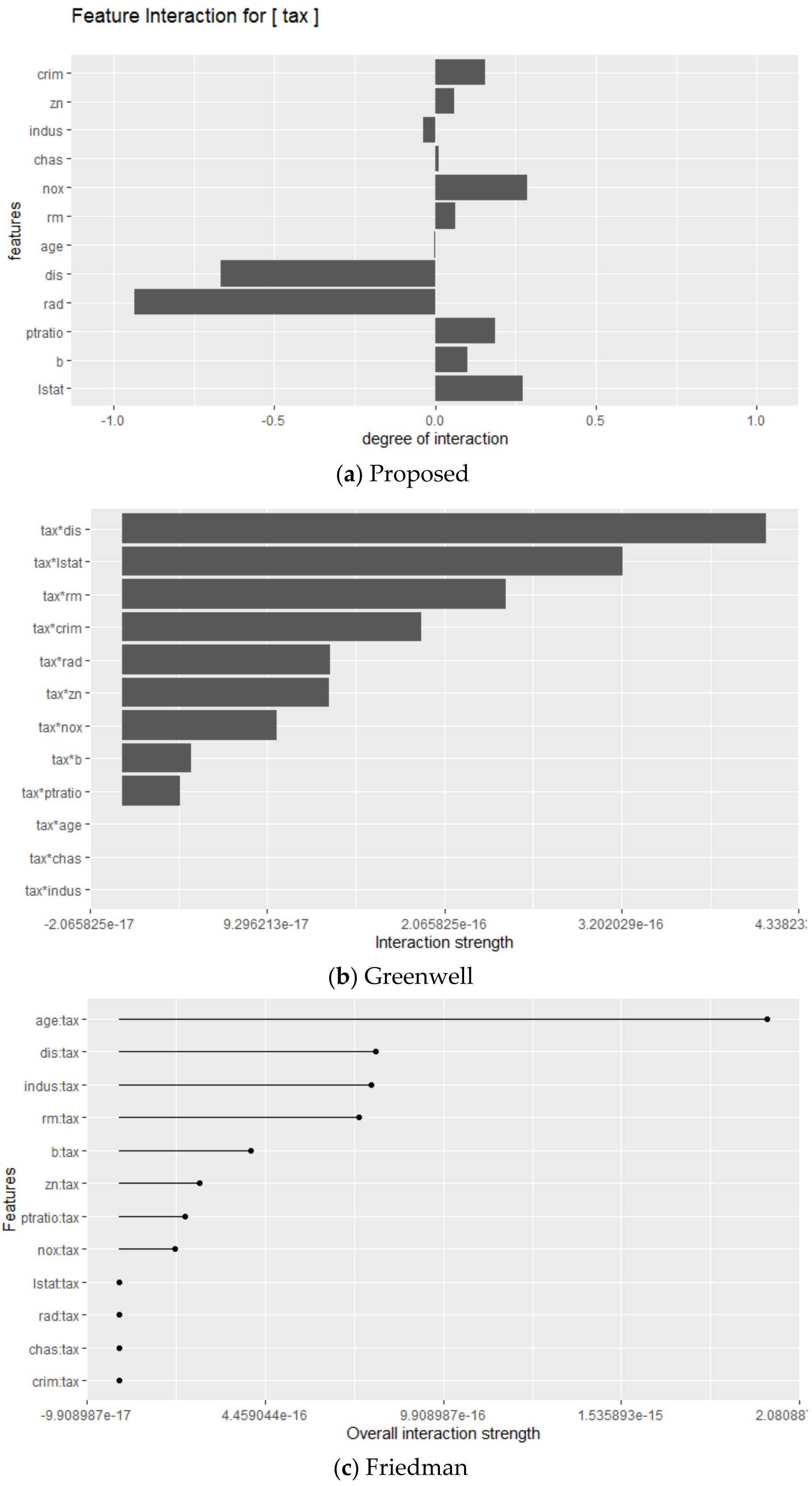
Figure 10. Top five features that interact positively/negatively with each other.

The source code of the implemented functions and the corresponding examples have been posted at "https://bitldku.github.io/home/sw/f_interaction.html\T1\textquotedblright.

4. Discussion

Feature interaction can be utilized in different applications owing to its alternate definition in this study compared to its prior definitions. In this section, the proposed method is compared with recent studies by Greenwell [8] and Friedman [17]. Figure 11 shows feature interaction graphs for variable "tax" and other variables in the BostonHousing dataset produced by the proposed method as well as those by Greenwell [8] and Friedman [17]. We can see that the graphs differ according to the definitions of feature interaction.

Greenwell and Friedman each focused on measuring feature interaction using partial dependency and H-statistics; the H-statistics were based on partial dependency. In these studies, feature interaction refers to the share of variances determined by the interaction [18]. In an interaction between features A and B, the partial dependency line in the PDP of feature A is changed/affected by that of feature B. In this study, feature interaction, which was measured using prediction performance, is defined as the variance of the prediction performance caused by the interaction.



(**a**) Proposed

(**b**) Greenwell

(**c**) Friedman

Figure 11. Feature interaction plots for variable tax and others.

Let us suppose variable A interacts with variable B in a specific model. In Greenwell's method, it measures the feature importance (defined as the variance of the partial dependence function) of one feature conditional on different, fixed points of the other feature [18]. If the variance is high, then the features interact with each other; if it is zero, they do not interact. Therefore, the interaction means that variable B influences the importance of variable A or vice versa. In other words, if variable B increases the importance of variable A, we can say B interacts with A. In Friedman's method, the interaction means that variable B influences partial dependency of variable A or vice versa. The meanings of feature interaction from Greenwell and Friedman are theoretically clear but difficult to intuitively understand. They can measure some influences between two features but do not show the meaning. In the proposed method, the interaction means that joining variable A and B increases or decreases prediction performance.

The partial dependency-based approach does not support the concept of "negative interaction" as the direction of interaction is not measured. In the proposed approach, negative interaction refers to the case wherein the merging of features reduces the prediction performance. In general, the addition of a new feature increases the prediction performance, although this might affect the existing prediction and reduce the prediction performance in some cases. This shows that feature selection should be carefully performed prior to fitting the prediction model. The proposed approach in this study facilitates the measurement of negative interactions in prediction models.

Greenwell's and Friedman's methods suffer from the time complexity involved in the computation of feature interactions [18]. The time complexity involved in the computation of the interaction between any two features is given by $O(N^2)$ and $O(2N^2)$, respectively, where N refers to the number of data instances (data points). Instance sampling techniques, such as the Monte Carlo method, have been adopted in most implementations to overcome the expensive time complexity. This approach reduces the computation time but also results in a loss of accuracy. In the proposed approach, a one-time permutation is first performed for two features, followed by a prediction test for N instances. This reduces the time complexity of the proposed approach to $O(N)$, thus achieving higher efficiency compared to previous approaches.

A majority of Friedman's method supports a standardized measure of feature interaction, which is based on an evaluation value between 0 and 1, to compare features across prediction models. However, it is difficult for users to interpret the absolute evaluation values as they are only capable of comparing their relative sizes. In the proposed approach, the evaluation values of feature interaction are not standardized. In the classification task, an evaluation value of 0.01 indicates that the interaction increases the prediction accuracy by 0.01. In the regression task, an evaluation value of 0.01 indicates that the interaction decreases the predicted RSME by 0.01. The evaluation value proposed in this paper was also compared across prediction models, which helps users clearly understand the feature interaction effects either in a specific model or across various models.

Table 3 summarizes the above discussion. Feature interaction can be defined variously and has differing characteristics according to the definition.

Table 3. Comparison of proposed method and previous works.

	Proposed	**Greenwell [8]**	**Friedman [17]**
R package	(self-implement)	vip [23]	iml [19]
Measure of interaction	prediction performance	partial dependency	H-statistics
Negative interaction	yes	no	no
Meaning of interaction	increases or decreases prediction performance	influences variable importance	influences partial dependency
Time complexity	$O(N)$	$O(N^2)$	$O(2N^2)$
[0,1] standardization	no	no	yes

5. Conclusions

This paper focused on two-way feature interaction. Although N-way feature interaction can be measured, it is relatively complex to interpret. The term "interaction" involves various aspects, which necessitates further research in feature interaction to identify new techniques for the interpretation of black box learning models. The feature interaction measure proposed in this study is simple to understand and can be quickly calculated. The proposed method can measure influences between features in a prediction model as a point of model performance (accuracy or error). It can capture negative interaction, whereas previous works captured only positive interaction. The proposed method is useful to understand the role of a feature and interaction between features in a prediction model. Furthermore, it may be used to improve model performance. For example, if a feature has low importance and a high negative interaction, we may consider removing it from the prediction model. This is a topic for further research.

Funding: This work was supported by the ICT & RND program of MIST/IITP [2018-0-00242, Development of AI ophthalmologic diagnosis and smart treatment platform based on big data].

Conflicts of Interest: The author declares no conflict of interest.

References

1. Panetta, K. Top Trends in the Gartner Hype Cycle for Emerging Technologies. Available online: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/ (accessed on 20 April 2019).
2. Schwartz, J. Machine Learning Is No Longer Just for Experts. Available online: https://hbr.org/2016/10/machine-learning-is-no-longer-just-for-experts (accessed on 20 April 2019).
3. Guidotti, R.; Monreale, A.; Turini, F.; Pedreschi, D.; Giannotti, F. A survey of methods for explaining black box models. *ACM Comput. Surv. CSUR* **2018**, *51*, 93. [CrossRef]
4. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Annu. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
5. Shapley, L.S. A value for n-person games. *Ann. Math. Stud.* **1953**, *20*, 307–317.
6. Friedman, J.H.; Popescu, B.E. Predictive learning via rule ensembles. *Ann. Appl. Stat. JSTOR* **2018**, *2*, 916–954. [CrossRef]
7. Giles, H. Discovering additive structure in black box functions. In Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 575–580.
8. Greenwell, B.M.; Boehmke, B.C.; McCarthy, A.J. A Simple and Effective Model-Based Variable Importance Measure. Available online: https://arxiv.org/abs/1805.04755 (accessed on 20 April 2019).
9. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
10. Fisher, A.; Rudin, C.; Dominici, F. Model Class Reliance: Variable Importance Measures for Any Machine Learning Model Class, from the 'Rashomon' Perspective. 2018. Available online: http://arxiv.org/abs/1801.01489 (accessed on 20 September 2018).
11. Hastie, T.; Tibshirani, R. Generalized Additive Models. In *Monographs on Statistics & Applied Probability*; Chapman & Hall/CRC: Boca Raton, FL, USA, 1990.
12. Loh, W. Regression trees with unbiased variable selection and interaction detection. *Stat. Sin.* **2002**, *12*, 361–386.
13. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 623–631.
14. Sorokina, D.; Caruana, R.; Riedewald, M.; Fink, D. Detecting statistical interactions with additive groves of trees. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1000–1007.
15. Deng, H. Interpreting tree ensembles with intrees. *Int. J. Data Sci. Anal.* **2019**, *7*, 277–287. [CrossRef]
16. Wright, M.N.; Ziegler, A.; König, I.R. Do little interactions get lost in dark random forests? *BMC Bioinform.* **2016**, *17*, 145. [CrossRef] [PubMed]
17. Boulesteix, A.L.; Janitza, S.; Hapfelmeier, A.; Van, S.K.; Strobl, C. Letter to the Editor: On the term 'interaction' and related phrases in the literature on random forests. *Brief. Bioinform.* **2015**, *16*, 338–345. [CrossRef] [PubMed]

18. Molnar, C. Interpretable Machine Learning. Available online: https://christophm.github.io/interpretable-ml-book/ (accessed on 10 April 2019).
19. Molnar, C. iml: Interpretable Machine Learning. Available online: https://github.com/christophM/iml (accessed on 15 April 2019).
20. Williams, C.K.; Engelhardt, A.; Cooper, T.; Mayer, Z.; Ziem, A.; Scrucca, L.; Tang, Y.; Candan, C.; Hunt, H.; Weston, S.; et al. Package 'Caret'. Available online: https://github.com/topepo/caret/ (accessed on 10 April 2019).
21. Newman, D.J.; Hettich, S.; Blake, C.L.; Merz, C.J. UCI Repository of Machine Learning Databases. Available online: http://www.ics.uci.edu/~{}mlearn/MLRepository.html (accessed on 15 April 2019).
22. Harrison, D.; Rubinfeld, D.L. Hedonic prices and the demand for clean air. *J. Environ. Econ Manag.* **1978**, *5*, 81–102. [CrossRef]
23. Greenwell, B.M.; Boehmke, B.C. Variable Importance Plots: An Introduction to Vip. Available online: https://cran.r-project.org/web/packages/vip/index.html (accessed on 10 September 2019).