Türkiye Klinikleri Biyoistatistik Dergisi
Turkiye Klinikleri Journal of Biostatistics Turkiye Klinikleri J Biostat. 2022;14(3):147-57

ORİJİNAL ARAŞTIRMA ORIGINAL RESEARCH DOI: 10.5336/biostatic.2022-91651

Evaluation of Traditional Machine Learning and Mixed Effect Machine Learning Model Performances by Simulation Study

Geleneksel Makine Öğrenmesi ve Karışık Etkili Makine Öğrenmesi Model Performanslarının Benzetim Çalışması ile Değerlendirilmesi

Ebru TURGAL[a]

[a]Department of Biostatistics, Ankara University Faculty of Medicine, Ankara, Türkiye

This study was presented as an oral presentation at 22nd National and 5th International Biostatistics Congress, October 28-30, 2021, Online.
This study was prepared based on the findings of Ebru TURGAL's thesis study titled "Evaluation of Traditional Machine Learning and Mixed Effect Machine Learning Model Performances by Simulation Study" (Ankara: Ankara University; 2022).

ABSTRACT Objective: The aim of this study is to examine the effectiveness of linear mixed-effects (LME) model, one of the traditional models used in the classification of clustered data, and mixed-effects machine learning models, which are the latest approaches. **Material and Methods:** For the simulation, various data sets were created with different number of groups (250, 500, 1000) and different sample sizes (5000, 10000, 15000). Within the scope of the simulation, LME model, mixed-effects random forest (MERF) and Gaussian process boosting (GPBoost) models were compared in terms of root mean square error (RMSE) on two functions. **Results:** When the error variance (EV) is 4 for the linear function, sample size is small and the number of groups is high, RMSE of MERF model is smaller. In all other scenarios, RMSE of the linear model was smaller and. In cases where EV for the nonlinear function is 1, the sample size is small and the number of groups is high, RMSE of MERF is smaller. In all other scenarios (while EV was 1), RMSE of GPBoost model was small, $p<0.05$ for the difference with LME, and $p>0.05$ for MERF. In cases where EV is 4 for the nonlinear function and the sample size and groups is high, RMSE of MERF is smaller. In all other scenarios (while EV was 4), RMSE of GPBoost model was smaller. **Conclusion:** As a conclusion, for a nonlinear function, GPBoost performed better than MERF and LME methods in terms of RMSE and time. However, when a linear function is considered, LME gives a better result.

Keywords: Grouped data; machine learning; random effects; simulation

ÖZET Amaç: Bu çalışmanın amacı, kümelenmiş verilerin sınıflamasında kullanılan geleneksel modellerden doğrusal karışık etkili [linear mixed-effects (LME)] model ile karışık etkili makine öğrenmesi modellerinin etkinliklerini incelemektir. **Gereç ve Yöntemler:** Benzetim tekniği için farklı küme sayılarında (250, 500, 1000) ve farklı örneklem büyüklüklerinde (5000, 10000, 15000) çeşitli veri setleri oluşturulmuştur. Benzetim çalışması kapsamında LME model, karışık etkili rastgele orman [mixed-effects random forest (MERF)] ve Gauss süreci boosting [Gaussian process boosting (GPBoost)] yöntemlerinin hata kareler ortalamasının karekökü [root mean square error (RMSE)] değeri bakımından karşılaştırılması 2 fonksiyon üzerinde gerçekleştirmiştir. **Bulgular:** Doğrusal fonksiyon için hata varyansı 4, örneklem sayısı az ve küme sayısı fazla olduğunda, doğrusal model yerine MERF modelinin RMSE daha küçük bulunmuştur. Bunun haricindeki tüm senaryolarda doğrusal modelin RMSE değerinin küçük olduğu görülmüştür. Doğrusal olmayan fonksiyon için hata varyansı 1, örneklem sayısının küçük ve küme sayısının yüksek olduğu durumlarda, MERF modelinin RMSE değeri daha küçük bulunmuştur. Bunun haricindeki tüm senaryolarda (hata varyansı 1 iken) GPBoost modelinin RMSE değerinin küçük, LME ile arasındaki fark için $p<0,05$, MERF ile ise farkın $p>0,05$ olduğu görülmüştür. Doğrusal olmayan fonksiyon için hata varyansı 4, örneklem ve küme sayısının yüksek olduğu durumlarda, MERF modeline ait RMSE daha küçük bulunmuştur. Bunun haricindeki tüm senaryolarda (hata varyansı 4 iken) GPBoost modelinin RMSE değeri küçük bulunmuştur. **Sonuç:** Sonuç olarak doğrusal olmayan bir fonksiyon için GPBoost; MERF ve LME yöntemine göre RMSE ve zaman açısından daha iyi bir performans göstermiştir. Ancak doğrusal bir fonksiyon ele alındığında LME daha iyi bir sonuç vermektedir.

Anahtar kelimeler: Kümelenmiş veri; makine öğrenmesi; rastgele etki; benzetim

Correspondence: Ebru TURGAL
Department of Biostatistics, Ankara University Faculty of Medicine, Ankara, Türkiye
E-mail: ebruturgal@gmail.com

Peer review under responsibility of Turkiye Klinikleri Journal of Biostatistics.

Received: 31 May 2022 ***Received in revised form:*** 17 Oct 2022 ***Accepted:*** 17 Oct 2022 ***Available online:*** 24 Nov 2022

The aim of this study is to examine the effectiveness of linear mixed-effects (LME) model, one of the traditional models used in the classification of clustered data, and mixed-effects machine learning models, which are the latest approaches in terms of root mean square error (RMSE) and time (second). Mentioned methods are mixed-effects random forest (MERF) and Gaussian process boosting (GPBoost) methods. They both are used for the localization of gene and protein markers for contributing additional solution in the field of medicine, more informative and promising diagnostic decision-making about disease pathogenesis and etiology. In the field of robotics; they perform learning with high speed and efficiency in motion planning, positioning, geolocation and mapping. In the field of sports; the prediction of NBA free agent player contract was modelled with MERF and this machine learning model will stay popular until a new model is built.

MATERIAL AND METHODS

GAUSSIAN PROCESS

Gaussian process is a flexible non-parametric function model that achieves state-of-the-art estimation accuracy and allows making probabilistic estimations.[1,2] This process, which is one of the probabilistic supervised machine learning methods, is used in both regression and classification problems.

Gaussian processes are used in areas such nonparametric regression, modelling of time series, spatial and spatio-temporal data, emulation of large computer experiments optimization of expensive black box functions and parameter tuning in machine learning models.[3-8]

In this approach, parameters are assumed to be independent, identically distributed random variables. Other supervised learning methods also learn exact values for each parameter. Gaussian process regression calculates the probability distribution over all acceptable functions that fit the data. First a priori point is determined, then the posterior point is calculated using the training data and compared with the estimated posterior at the respective points. In this method, result of $\sum(y - \hat{y})^2$ equation should be minimal.[9] Because it means that the difference between predicted and true values is minimal.

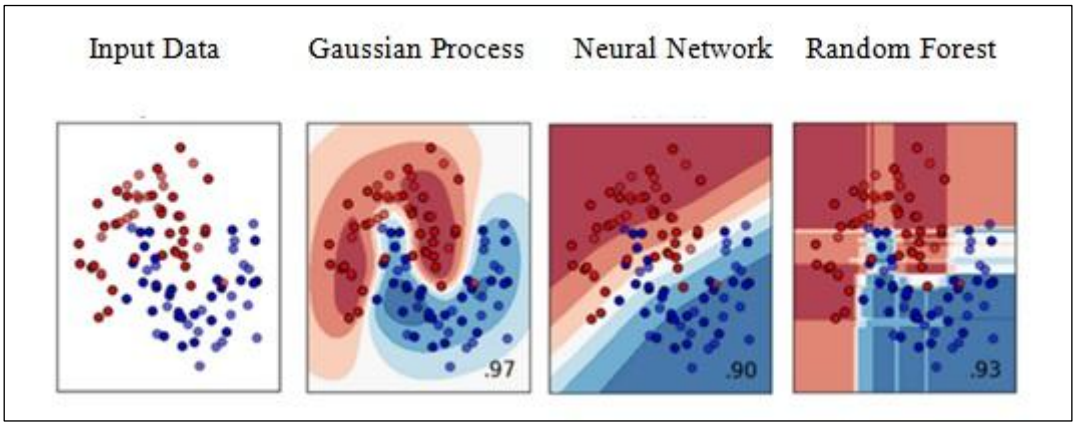

FIGURE 1: Classifier comparison outputs.[10]

Figure 1 shows the classification functions learned by the different methods used to separate the blue and red dots. Neural network and random forest, which are widely used and powerful methods, produce predictions away from the training data. The Gaussian process obtains the model output with greater accuracy, which is particularly important in authentication and security-critical uses.[1]

The Gaussian process is a non-parametric Bayesian approach to regression used to approximate functions.[9] The Bayesian approach determines a probability distribution over possible functions.[10] The algorithm

for this process is a powerful algorithm for modelling nonlinear relationships between the pairs of random variables. It defines a distribution over functions that can be applied to demonstrate uncertainty about the actual functional relationship.[11] In addition, the Gaussian process is widely used in motion planning, positioning, localization and mapping in the field of robotics.[12,13]

Gaussian process regression is used in areas that need to be modelled on people's preferences (such as understanding user preferences). Today, it is used in the modelling of dynamic animations such as human walking, dinosaur walking, fire, and explosion. After the low-resolution animations are shown to people and asked to rate them, the model parameters are finalized by scoring the people's perception of reality. The Gaussian regression process models people's preferences as a function. Since the process is started as Bayesian, the model that starts with preliminary information at first becomes close to reality in the end. Thus, a high cost task is solved with a low processor cost.

Gaussian processes are used not only in regression problems, but also in classification problems. An example is Google's classification problems. When a picture is searched with the word 'cat' on Google, the search engine returns many pictures that it thinks are cats. When clicked, the photo that most resembles a cat, and the photo that the user prefers to print first, is generally accepted as the photo that most resembles a cat. The most clicked with a high voting rate categorizes it as “cat”. Then, by using this prior information, the search engine brings forward the photos that look more like cats in the cat search of later users by increasing the estimation performance in the “cat” classification problem.

GPBoost

The model is trained using the GPBoost algorithm. This takes place by training the covariance parameters of random effects and the mean F(X) function with a tree ensemble model. The algorithm used is a boosting algorithm that iteratively learns the covariance parameters and adds a tree to the tree ensemble using a gradient or Newton acceleration step. Covariance parameters can be learned using (Nesterov acceleration) gradient descent or Fisher scoring.[14]

Clustered data structure is a concept that emerges when data comes together under groups with certain characteristics.[15] Patients in different hospitals, students in schools, teeth in the mouth can be given as examples of this data structure. Although these related data structures (panel, longitudinal and clustered data) are defined as special parts of multivariate designs, they contain many differences that completely affect the way of analysis.

Clustered data can be modelled as follows;

1. Group structure can be ignored.

2. Each group (e.g. each student) can be modelled separately.

3. The grouping variable (e.g. student or customer ID) is included in the model and treated as a categorical variable. While this is a viable approach, it has some disadvantages. Mostly, the number of measurements per group (e.g., number of tests per student) is small, and the different groups are large (e.g., students, etc.). In this case, the model needs to learn many parameters (one for each group) based on less data, which can make the learning inefficient. Also, large numbers of categorical variables can be problematic for trees.

4. The grouping variable is treated as random effects, and the Gaussian process and the mixed-effects model approach can be used.[14]

Briefly, this approach proposes to model the mean function and the predicted parameters of the covariance structure of random effects together with the mean function, with a population of basic learners such as regression trees that learn incrementally using boosting.[16]

In the machine learning literature, F(X)=0 is assumed in mixed-effects models with the Gaussian process, while the priori distribution function is assumed to be as in Equation 1 in mixed-effects models with the Gaussian process.

$F(X) = X^T\beta, \beta\epsilon\mathbb{R}^P,$ vector of covariate (Equation 1)

The purpose of this approach is to relax the linearity assumption, or the zero-prior mean assumption.

GPBoost is built on the popular tree-boosting models, which have state-of-the-art predictive accuracy, such as eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM).[17,18] The nonlinear function, F, in GPBoost is built on LightGBM, but trained with the GPBoost algorithm nto learn the covariance parameters of the random effects and the nonlinear function F(X) using a LightGBM tree ensemble.[19]

SIMULATION STUDY

Data generation was performed in different scenarios according to sample size, number of groups, random effect variance and error variance. The sample sizes for the simulation technique are 5000, 10000 and 15000; number of groups was taken as 250, 500, 1000.

Within the scope of the simulation study, the LME model, MERF and GPBoost methods were compared in terms of RMSE value on two functions. The first of these functions is a linear function and is shown in Equation 2. In addition, a single grouping variable was used. However, hierarchically nested random effects and crossed random effects can be used. Random effect variance of 1, error variance of 1 and 4 were taken separately in the study. Each scenario was repeated 100 times. A linear function is included to observe the comparison between the mixed-effects machine learning model and a LME model. This function is shown in Equation 2. In addition, the nonlinear function "friedman3" is also included in the simulation study. "friedman3" function was first introduced by Friedman (1991) and is frequently used to compare nonparametric regression models (Equation 3).[20] This function is a nonlinear function containing 4 independent variables and since it has a multidimensional structure, it is generally tried to obtain the closest estimation value to the y response variable by using this function within the scope of machine learning.

$y\,(x)$ = C* x + 2 (Equation 2)

$y\,(x)$ = C* arctan $\left(\frac{x_1x_2-\frac{1}{x_1x_3}}{x_0}\right)$ (Equation 3)

The constant C is chosen so that the variance of F(X) is equal to 1, as can be seen in previous equations, namely F(X) has the same power as the random effects.

$y = F(X) + Zb + \varepsilon, b \sim N(0, \sum), \varepsilon_i \sim N(0, \sigma^2 I_n)$ (Equation 4)

A single grouping variable was used. However, random effects can also be used, including hierarchically clustered, nested, crossed, and random coefficient effects.

MODEL PARAMETERS FOR GPBoost

Max_Depth: Tree based algorithms have 4 different components: root, internal, leaf nodes and brances. maxdepth is the value of the tree's branches (edges) extending downwards. Any point where a decision must be made is known as a decision node. Branches are the branches that extend from a decision node, and each branch represents one of the options or possible courses of action that are currently available. In other words, *max_depth* is the depth of the tree. Depth of the tree is the total number of edges from root to leaf in the longest part. It should be optimized to avoid over-learning. Too much branching causes over-learning and too little branching causes under-learning. *maxdepth* tells to split the tree to 3 times. 0 indicates no limit on depth. In the current study it is taken as 6.

Learning_rate: A value between 0-1 for scaling trees. A smaller value helps better predictive power. But, it will increase learning time and the possibility of over-learning. The default parameter is taken as 0.1.

Iterations: It shows the number of trees to be created. It is also used with the names "*num_boost_round*", "*n_estimators*", "*num_trees*", and "*num_iterations*" in different algorithms. If it is taken too small, it can cause under-learning, and too much can cause over-learning. In addition, the increase in the number increases the training time. The default parameter is taken as 100.

Early_stopping_rounds: It is the parameter used to prevent over-learning. After finding the most suitable step, number of trials should be specified. With the value of 100, the model performs 100 more iterations after the optimal, and the model stops learning even if the target parameters are not captured. For example, if the number of iterations is 2000 in the first parameter, if the optimal moment is reached in the 1000th iteration, the model will stop there. The default parameter is taken as 5.

Verbosity: In each iteration, the model's learning status, total time and remaining time are output. This output takes up too much space on the screen in case of multiple iterations and does not provide enough information to be worth it. The default parameter is set to 0.

Feature_fraction: If less than 1.0, it randomly selects a subset of features (parameters) at each iteration (tree). For example, if taken as 0.8, the algorithm will select 80% of the parameters before training each tree. The default parameter is 1.0.

Subsample: The subsample ratio of the training sample. It checks the samples given to the trees. For example, if set to 0.5 it means allowing the algorithm to randomly sample half of the training data before growing the trees, which will prevent over-learning. The default parameter is 1.0.

In order to make an objective comparison of the decision tree algorithm and boosting algorithms used, the basic parameters used in the algorithms were entered the same for each algorithm. Parameters with the same values are respectively; "*n_estimators*" are "*learning_rate*" and "*max_depth*". A high "*max_depth*" value creates a more complex model and increases the likelihood of overfitting. In addition, the cross validation value operation was performed by determining the K-fold value as 4. The cross validation value process was used in all models in this study with the same parameter values. On the other hand, "n_jobs" parameter ensures that the calculation process is done using parallel processors, so that the processes are carried out faster. Taking the value of -1 for this parameter ensures that all CPUs are used.[21]

TUNING PARAMETERS FOR GPBoost

Tuning parameters were chosen using 4-fold cross validation as seen on Figure 2 on the training data with RMSE as a criterion and ignoring random effects for GPBoost model predictions.

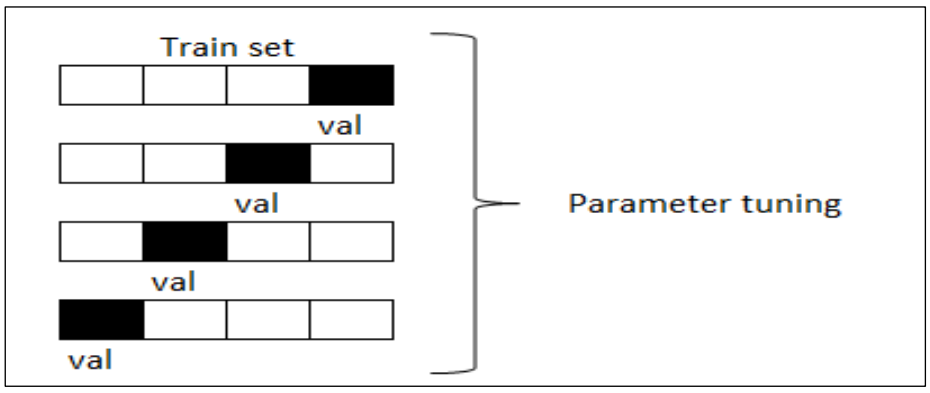

FIGURE 2: Parameter tuning for the training set using 4-fold cross validation.[21]

MODEL PARAMETERS FOR MERF

merf Python package (version 0.3) is used for the MERF algorithm. A maximum tree depth limit was not set and the number of trees was set to 300. These are the default values of the MERF package.[22] For the MERF algorithm; we choose the proportion of variables considered for making splits ∈ {0.5, 0.75, 1}. These are the default values of the MERF package and have also been used in Hajjem et al.[12]

SYSTEM CAPACITY

All remaining calculations are performed on a personal computer with Intel(R) Xeon(R) Gold 6136 3 GHz 16-core processor and 256 GB random access memory. In addition, Python Jupyter Notebook 6.4.8 was used in the analysis in the study.

RESULTS

A total of 108 different scenarios were created according to the cases where the sample size was 5000, 10000 and 15000, the number of groups was 250, 500 and 1000, the random effect variance was 1 and the error variance was 1 and 4. Results for all scenarios RMSE, standard deviation of RMSE and time (second) criterias are tabulated in Table 1, Table 2, Table 3, Table 4, Table 5 and Table 6 and the order of importance for the main feature selection of the "friedman3" model is made on the figures using the SHApley Additive exPlanations (SHAP) technique.

TABLE 1: Results for the grouped random effects model and the mean F='linear (for sample size=5000).

				Linear ME			MERF			GPBoost			p value	
Sample size	Number of groups	σ_b^2	σ_ε^2	RMSE	SD	Time (s)	RMSE	SD	Time (s)	RMSE	SD	Time (s)	Linear ME-GPBoost	MERF-GPBoost
5000	250	1	1	**1.254**	0.0212	<0.001	1.260	0.0212	172.01	1.264	0.0209	0.813	8.67E-59	9.41E-38
	500			**1.246**	0.0178	0.016	1.252	0.0179	188.58	1.253	0.0179	0.053	1.55E-62	1.89E-43
	1000			**1.264**	0.0134	<0.001	1.268	0.0135	283.73	1.271	0.0131	0.228	1.98E-66	2.42E-38
5000	250	1	4	**2.139**	0.0226	<0.001	2.149	0.0228	131.162	2.161	0.0225	0.098	9.63E-58	1.47E-36
	500			**2.134**	0.0310	<0.001	2.147	0.0235	186.91	2.145	0.0254	0.094	0.0105	2.86E-10
	1000			2.252	0.0245	0.016	**2.183**	0.0209	283.36	2.187	0.0321	0.141	0.0173	5.83E-40

σ_b^2 : Random effects variance; σ_ε^2 : Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: Standard deviation.

TABLE 2: Results for the grouped random effects model and the mean F='linear (for sample size=10000).

				Linear ME			MERF			GPBoost			p value	
Sample size	Number of groups	σ_b^2	σ_ε^2	RMSE	SD	Time (s)	RMSE	SD	Time (s)	RMSE	SD	Time (s)	Linear ME-GPBoost	MERF-GPBoost
10000	250	1	1	**1.250**	0.0217	<0.001	1.252	0.0216	154.57	1.254	0.0217	0.047	1.85E-68	1.79E-44
	500			**1.246**	0.0157	<0.001	1.249	0.0158	211.10	1.251	0.0157	0.098	2.80E-63	3.75E-43
	1000			**1.244**	0.0109	0.016	1.246	0.0108	329.77	1.248	0.0109	0.047	5.93E-62	2.79E-41
10000	250	1	4	**2.142**	0.0184	<0.001	2.144	0.0185	153.73	2.147	0.0186	0.173	1.65E-67	4.32E-40
	500			**2.135**	0.0162	0.016	2.140	0.0163	212.83	2.144	0.0163	0.176	1.04E-60	3.14E-40
	1000			**2.149**	0.0166	<0.001	2.152	0.0143	319.77	2.156	0.0143	0.141	1.61E-14	1.34E-34

σ_b^2 : Random effects variance; σ_ε^2 : Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: Standard deviation.

TABLE 3: Results for the grouped random effects model and the mean F='linear (for sample size=15000).

				Linear ME			MERF			GPBoost			p value	
Sample size	**Number of groups**	σ_b^2	σ_ε^2	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**Linear ME-GPBoost**	**MERF-GPBoost**
15000	250	1	1	**1.242**	0.0193	<0.001	1.244	0.0192	225.95	1.246	0.0192	0.113	9.37E-68	1.16E-42
	500			**1.253**	0.0144	<0.001	1.254	0.0143	216.71	1.255	0.0144	0.109	1.40E-63	8.49E-38
	1000			**1.243**	0.0106	0.016	1.245	0.0106	328.19	1.246	0.0105	0.098	6.23E-65	2.04E-43
15000	250	1	4	**2.127**	0.0152	0.016	2.132	0.0152	228.20	2.134	0.0152	0.237	1.73E-67	5.23E-43
	500			**2.141**	0.0139	<0.001	2.143	0.0138	216.34	2.146	0.0142	0.191	1.28E-60	1.20E-38
	1000			**2.144**	0.0124	0.016	2.147	0.0125	327.82	2.150	0.0124	0.238	1.24E-69	1.53E-45

σ_b^2 : Random effects variance; σ_ε^2 : Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: Standard deviation.

TABLE 4: Results for the grouped random effects model and the mean F='friedman3'. (for sample size=5000).

				Linear ME			MERF			GPBoost			p value	
Sample size	**Number of groups**	σ_b^2	σ_ε^2	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**Linear ME-GPBoost**	**MERF-GPBoost**
5000	250	1	1	1.395	0.0214	<0.001	1.256	0.0237	133.599	**1.251**	0.0229	0.063	9.66E-118	0.05130
	500			1.391	0.0187	<0.001	1.248	0.0171	197.464	**1.248**	0.0167	0.083	4.37E-111	0.00032
	1000			1.405	0.0163	<0.001	**1.255**	0.0154	303.210	1.258	0.0149	0.063	1.19E-116	0.30796
5000	250	1	4	2.221	0.0241	<0.001	**2.147**	0.0254	133.345	2.150	0.0242	0.129	5.82E-94	2.22E-34
	500			2.222	0.0345	<0.001	**2.140**	0.0222	200.187	2.145	0.0260	0.123	3.03E-53	9.10E-10
	1000			2.275	0.0252	<0.001	**2.152**	0.0223	315.551	2.214	0.0297	0.121	1.95E-53	2.57E-48

σ_b^2: Random effects variance, σ_ε^2: Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: SS: Standard deviation.

TABLE 5: Results for the grouped random effects model and the mean F='friedman3'. (for sample size=10000).

				Linear ME			MERF			GPBoost			p value	
Sample size	**Number of groups**	σ_b^2	σ_ε^2	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**Linear ME-GPBoost**	**MERF-GPBoost**
10000	250	1	1	1.395	0.0194	<0.001	1.246	0.0215	161.466	**1.246**	0.0211	0.083	6.27E-129	9.49E-15
	500			1.399	0.0150	<0.001	1.243	0.0143	216.028	**1.238**	0.0144	0.079	1.65E-128	3.74E-29
	1000			1.414	0.0126	0.008	1.256	0.0128	358.004	**1.251**	0.0126	0.113	2.67E-133	3.08E-32
10000	250	1	4	2.238	0.0187	<0.001	**2.156**	0.0192	157.910	2.161	0.0190	0.145	2.08E-111	6.71E-30
	500			2.258	0.0180	0.016	**2.162**	0.0166	213.937	2.164	0.0166	0.145	1.49E-105	5.82E-27
	1000			2.280	0.0258	0.016	**2.181**	0.0172	473.814	2.187	0.0217	0.177	2.023E-70	1.65E-05

σ_b^2: Random effects variance, σ_ε^2: Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: Standard deviation.

TABLE 6: Results for the grouped random effects model and the mean F='friedman3'. (for sample size=15000).

				Linear ME			MERF			GPBoost			p value	
Sample size	**Number of groups**	σ_b^2	σ_ε^2	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**RMSE**	**SD**	**Time (s)**	**Linear ME-GPBoost**	**MERF-GPBoost**
15000	250	1	1	1.376	0.0170	0.016	1.241	0.0192	215.126	**1.229**	0.0189	0.114	6.00E-145	1.09E-27
	500			1.382	0.0132	<0.001	1.235	0.0142	222.246	**1.229**	0.0143	0.129	4.70E-141	1.51E-47
	1000			1.385	0.0108	0.016	1.248	0.0117	335.205	**1.242**	0.0120	0.141	1.62E-145	1.38E-53
15000	250	1	4	2.220	0.0168	<0.001	**2.141**	0.0170	220.922	2.142	0.0171	0.206	3.09E-124	1.74E-20
	500			2.234	0.0143	0.016	**2.148**	0.0136	221.315	2.151	0.0138	0.240	1.06E-119	5.40E-20
	1000			2.238	0.0130	<0.001	2.162	0.0136	344.900	**2.160**	0.0140	0.252	3.84E-124	8.66E-26

σ_b^2: Random effects variance, σ_ε^2: Error variance; ME: Mixed-effects; MERF: Mixed-effects random forest; GPBoost: Gaussian process boosting; RMSE: Root mean square error; SD: Standard deviation.

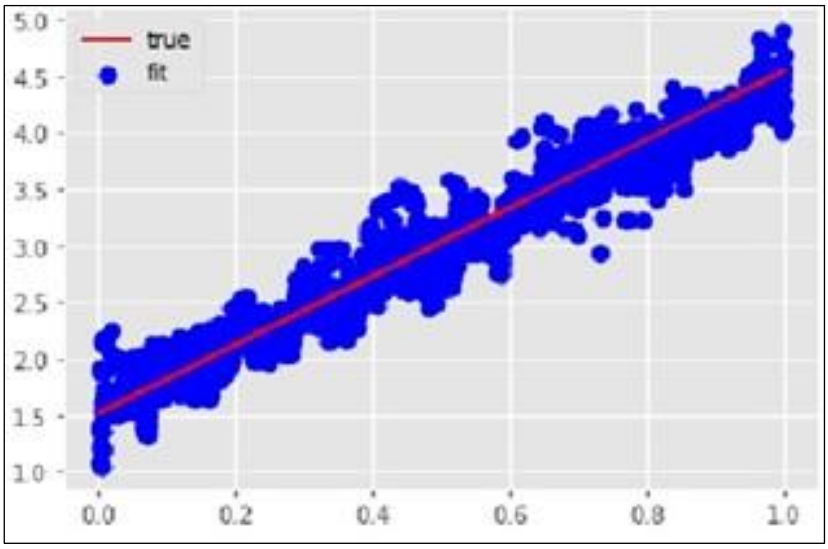

FIGURE 3: Comparison of actual and estimated fixed effects for the 'linear' function ($\sigma_b^2 = 1$ and $\sigma_\varepsilon^2 = 1$, n=5000, number of groups m=250).

Figure 3 shows a comparison of the true and fitted fixed effects for the 'linear' function ($\sigma_b^2 = 1$ and $\sigma_\varepsilon^2 = 1$, n=5000, number of groups m=250)

SHAP TECHNIQUE

The SHAP technique was used to evaluate the importance of each feature in estimating the response variable.[23] In the SHAP technique, the model can be interpreted based on the Shapley values that explain the contribution of each feature to the prediction.[24] According to Figure 4, it is seen that the "variable 1" attribute contributes the most to the prediction model in determining the *y* response variable.

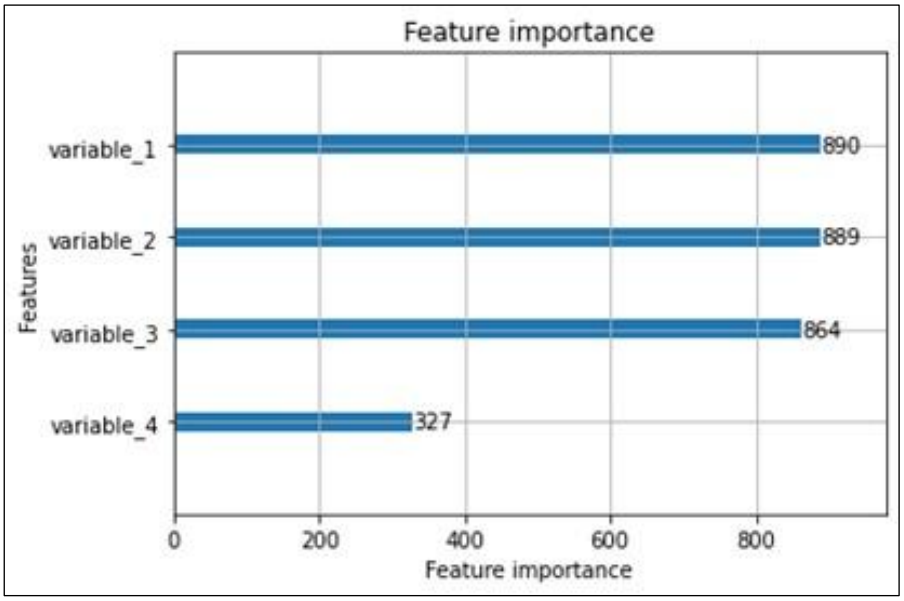

FIGURE 4: SHApley Additive exPlanations values for the "friedman3" function (for Equation 3).

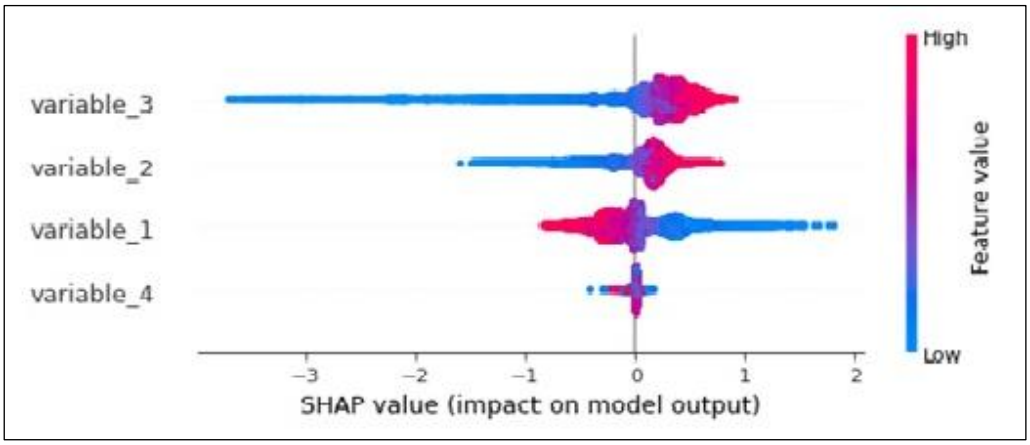

FIGURE 5: "friedman3" function (for Equation 3) SHAP summary graph.

SHAP: SHApley additive explanations.

A variable has a high SHAP value (and because it is in the negative direction) means that it is associated with high and positive values on its estimation (Figure 5). Figure 5 shows the order of importance of 4 variables. The SHAP summary graph shows the most important features and their impact on the dataset. This graph shows that there is a positive correlation between "variable 2" and "variable 3" and the response variable when the "friedman3" function is used. Also it shows a negative relationship between "variable 1" and the response variable.

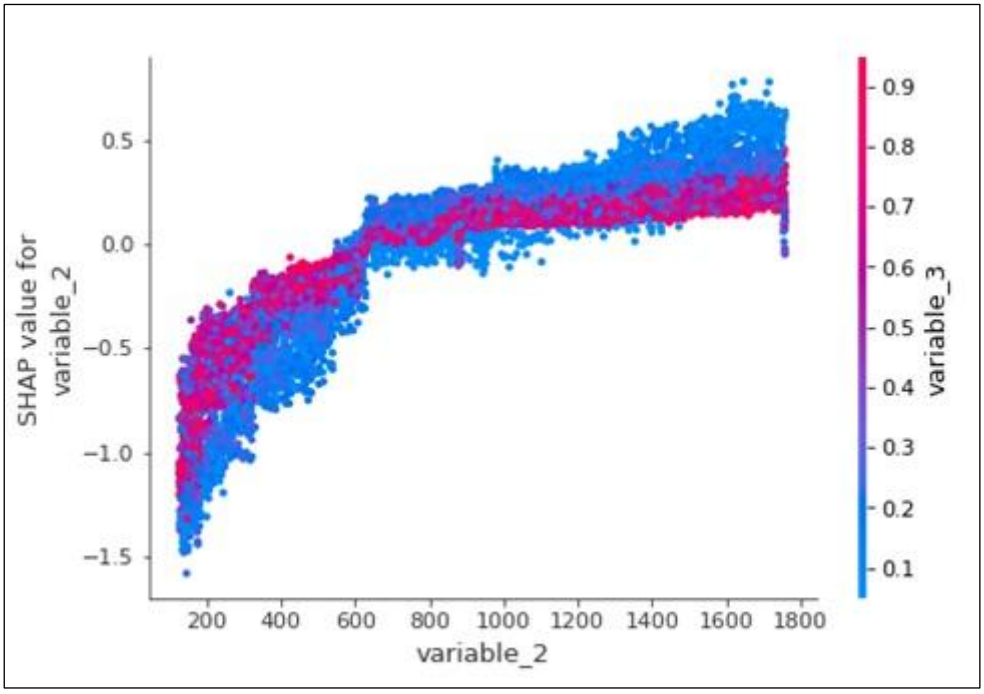

FIGURE 6: "friedman3" function (for Equation 3) SHAP interaction graph.

SHAP: SHApley additive explanations.

Figure 6 and shows that "variable 2" interacts frequently with "variable 3". In this drawing, the gradient color of each point corresponds to the original value of the number of "variable 3" from low (blue) to high (red).

DISCUSSION

It is recommended to use generalized mixed-effects models to relax the linearity assumption in mixed-effects models. This may cause the model to be built incorrectly. For this reason, clustered or grouped random effects models for longitudinal data or non-parametric, machine learning-based approaches have been proposed. The MERF model was used by Manifold AI,[25] and GPBoost model was used by Sigrist.[14] GPBoost machine learning algorithm is an effective method for not only relaxing the linearity assumption in Gaussian process and mixed-effects models, but also the independence assumption for boosting. If the mixed-effects machine learning model contains a linear function, LME method will often be more efficient than the GPBoost and MERF methods, both in terms of RMSE and time (s), but if it contains a nonlinear function, the performance of the GPBoost method is generally better compared to other models.

Considering the results obtained, when the error variance is 4 for the linear function, the sample size is small and the number of groups is high, the RMSE of the MERF model is smaller than the linear model, and this difference is statistically significant. In all other scenarios, it is seen that the RMSE of the linear model is small and this difference is statistically significant. Sigrist, on the other hand, indicated that the RMSE of the LME model to be smaller and this difference statistically significant in each simulation model where the number of groups is small and the sample size is low.[14]

In cases where the error variance for the nonlinear function is 1, the sample size is low and the number of groups is high, the RMSE of the MERF model is smaller instead of the GPBoost model, and this difference is statistically significant. In all other scenarios (while the error variance is 1), it is seen that the RMSE of the GPBoost model is small, the difference between LME is statistically significant, and the difference between MERF is not statistically significant. Sigrist, on the other hand, indicated that the RMSE of the GPBoost model to be small and this difference statistically significant in every simulation model where the number of groups and the sample size is low.[14]

In cases where the error variance is 4 for the nonlinear function and the sample size and groups is high, the RMSE of the MERF model is smaller instead of the GPBoost model, and this difference is statistically significant. In all other scenarios (when the error variance is 4), the RMSE of the GPBoost model is small and this difference is statistically significant.

In order to be applied to real life data, there is a need in the literature to carry out simulation studies and studies in which sample size, number of groups and error variances are handled in different ways.

CONCLUSION

In the simulation step of the study, for a nonlinear function, GPBoost performed better than MERF and LME methods in terms of RMSE and computational time. However, when a linear function is considered, LME gives better results. p values were also calculated from t-tests for dependent groups comparing the GPBoost algorithm with other approaches in terms of RMSE. The results showed that the difference between the methods was statistically significant ($p<0.001$). In terms of computational time, it is observed that the speed of the MERF algorithm is very slow compared to the others, which is thought to be due to the fact that it has not a properly defined expectation-maximization algorithm.[14]

Mixed-effects models with the Gaussian process method, in the field of medicine; is used for the localization of gene and protein markers for contributing additional solution, more informative and promising diagnostic decision-making about disease pathogenesis and etiology. In the field of sports; the prediction of NBA free agent player contract was modelled with MERF and this machine learning model will stay popular until a new model is built.[26]

Acknowledgment

Special thanks to Prof. Dr. Atilla Halil ELHAN and Assoc. Dr. Beyza DOĞANAY ERDOĞAN for their support in my work.

Source of Finance

During this study, no financial or spiritual support was received neither from any pharmaceutical company that has a direct connection with the research subject, nor from a company that provides or produces medical instruments and materials which may negatively affect the evaluation process of this study.

Conflict of Interest

No conflicts of interest between the authors and/or family members of the scientific and medical committee members or members of the potential conflicts of interest, counseling, expertise, working conditions, share holding and similar situations in any firm.

Authorship Contributions

This study is entirely author's own work and no other author contribution.

REFERENCES

1. Knagg O. An Intuitive Guide to Gaussian Processes. Towards Data Science. 2019. Cited: December 10, 2020. Available from: [Link]
2. Gneiting T, Balabdaoui F, Raftery AE. Probabilistic forecasts, calibration and sharpness. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2007;69(2):243-68. [Crossref]
3. Shumway RH, Stoffer DS, Stoffer DS. Time Series Analysis and its Applications. 3rd ed. New York: Springer; 2000. [Crossref]
4. Banerjee S, Carlin BP, Gelfand AE. Hierarchical Modeling and Analysis for Spatial Data. 1st ed. New York: Chapman and Hall/CRC; 2003. [Crossref]
5. Cressie N, Wikle CK. Statistics for Spatio-Temporal Data. 1st ed. Hoboken, NJ: John Wiley & Sons; 2015.
6. Kennedy MC, O'Hagan A. Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2001;63(3):425-64. [Crossref]
7. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. Journal of Global Optimization. 1998:13(4):455-92. [Crossref]
8. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems. 2012;25. [Crossref]
9. Ateş B. Gemi yapılarında gerilme yığılması öngörülerinin kaba ağ yapısı ve makine öğrenmesi ile gerçekleştirilmesi [Yüksek lisans tezi]. İstanbul: İstanbul Teknik Üniversitesi; 2020. Erişim tarihi: 16.06.2021 [Link]
10. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in python. Journal of Machine Learning Research. 2011:12:2825-30. [Crossref]
11. Williams CK, Rasmussen CE. Gaussian Processes for Machine Learning. 1st ed. Cambridge, MA: MIT Press; 2006. [Crossref]
12. Hajjem A, Larocque D, Bellavance F. Generalized mixed effects regression trees. Statistics & Probability Letters. 2017;126:114-8. [Crossref]
13. Yangın G. XGboost ve karar ağacı tabanlı algoritmaların diyabet veri setleri üzerine uygulaması [Yüksek lisans tezi]. İstanbul: Mimar Sinan Güzel Sanatlar Üniversitesi; 2019. Erişim tarihi: 10.07.2021 [Link]
14. Sigrist F. Gaussian process boosting. arXiv. 2020:1-42. [Crossref]
15. Song XK, Song PXK. Correlated Data Analysis: Modeling, Analytics, and Applications. 1st ed. New York: Springer Science & Business Media; 2007.
16. Li B, Friedman J, Olshen R, Stone C. Classification and regression trees (CART). Biometrics. 1984;40(3):358-61. [Crossref]
17. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016;785-94. [Crossref]
18. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: a highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems. 2017;30:3146-54. [Crossref]
19. Zhou F, Alsaid A, Blommer M, Curry R, Swaminathan R, Kochhar D, et al. Predicting driver fatigue in monotonous automated driving with explanation using GPBoost and SHAP. International Journal of Human-Computer Interaction. 2022;38(8):719-29. [Crossref]
20. Friedman JH. Multivariate adaptive regression splines. The Annals of Statistics. 1991;19(1):1-67. [Crossref]
21. Coşkun K. Ağ saldırılarının sınıflandırılmasında karar ağaçlarına dayalı arttırma (boosting) algoritmalarının karşılaştırılması [Yüksek lisans tezi]. Muğla: Muğla Sıtkı Koçman Üniversitesi; 2020. Erişim tarihi: 16.09.2021 [Link]
22. Hajjem A, Bellavance F, Larocque D. Mixed-effects random forest for clustered data. Journal of Statistical Computation and Simulation. 2014;84(6):1313-28. [Crossref]
23. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems. 2017;1-10. [Crossref]
24. Rodríguez-Pérez R, Bajorath J. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. J Comput Aided Mol Des. 2020;34(10):1013-26. [Crossref] [PubMed] [PMC]
25. Manifold AI [Internet]. [Cited: August 1, 2019]. Mixed effects random forest (Python). 2014. Available from: [Link]
26. Natarajan SS. A Mixed Effects Modeling Approach to Predicting NBA Free Agency. 2019. Cited: August 07, 2020. Available from: [Link]