

Received November 19, 2020, accepted November 26, 2020, date of publication November 30, 2020, date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041367

A Comparison Framework of Machine Learning Algorithms for Mixed-Type Variables Datasets: A Case Study on Tire-Performances Prediction

LEONARDO GUTIÉRREZ-GÓMEZ¹, FRANK PETRY², AND DJAMEL KHADRAOUI¹

¹Luxembourg Institute of Science and Technology (LIST), 4362 Esch-sur-Alzette, Luxembourg

²Goodyear Innovation Center* Luxembourg, 7750 Colmar-Berg, Luxembourg

Corresponding author: Leonardo Gutiérrez-Gómez (leonardo.gutierrez@list.lu)

ABSTRACT Many engineering applications in the automotive, aeronautic, rubber, mechanics, and manufacturing industries collect multiple datasets measuring physical relations between input variables and performances for modeling purposes. The challenge relies on that such data is often highly dimensional, non-linear and contain mixed variables, i.e., numerical and categorical features, requiring specific algorithms and encoding schemes to perform regression task efficiently. Moreover, defining an appropriated similarity criterion for mixed-type data is a non-trivial task, especially when it is meant to be used in regression problems. This paper discusses the use of different machine learning algorithms for regression problems, involving *mixed-type* variables across *multiple* datasets. We use tire-related datasets as a case study to perform a rigorous, statistically founded comparison of different machine learning algorithms with encoding schemes to handle mixed variables in the prediction of tire-performances across multiple tire-related datasets. Friedman's statistic and Nemenyi post-hoc tests are used to test the significance of performance differences between techniques and encoding strategies. Our contributions come as a series of recommendations for handling efficiently mixed-type variables while achieving high performances on regression tasks over multiple datasets. Furthermore, we provide a flexible and efficient similarity function between tires useful for tire comparison, prediction, and retrieval tasks.


INDEX TERMS Mixed-type variables, categorical encoding, Friedman Nemenyi, regression algorithms.

I. INTRODUCTION

Machine learning (ML) in engineering applications has grown in popularity during the last decades [1]–[5]. Many industrial applications use ML tools to build regression models for product design, performance optimization, variable design, fault detection, quality assessment, and others. For instance, the rubber industry, [6] uses non-linear least squares to estimate the tire-road friction coefficient for tire design. In automotive design, [7] employs support vector regression in structural optimization to vehicle crashworthiness design. More recently [8] performed thermodynamics compressor performance modeling for engine design with neural networks and non-linear support vector regression. A common characteristic of engineering data is its tabular-like structure, where rows represent data examples, which are themselves described as a mixture of numerical and categorical, i.e., *mixed-type* variables. For instance, in car crashworthiness

design [9], vehicle structures must be designed to absorb crash energy through structural deformation as much as possible and attenuate the impact force to lower levels when impact occurs. The design variables are thickness related (measured in mm) and steel hardness types. For instance, the B-pillar inner, reinforce, the floor side and door beltline, are all *numerical* variables. However material design variables associated to the steel hardness, i.e., meal, medium, or high strength steel are *categorical* variables as well as material types such as iron, aluminium, plastic steel, glass, rubber and copper [9]. Besides, engineering problems often involve *multiple* inter-related datasets describing complementary performance measurements of the system of interest (see our case study). Therefore, assessing the overall performance of a ML algorithm in the regression tasks across multiple engineering datasets, is a challenging task in particular when the data contains *mixed-type* features.

From the data modelling perspective, dealing with mixed-variables may be problematic for many ML algorithms. As categorical variables do not have explicit ordering

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry .

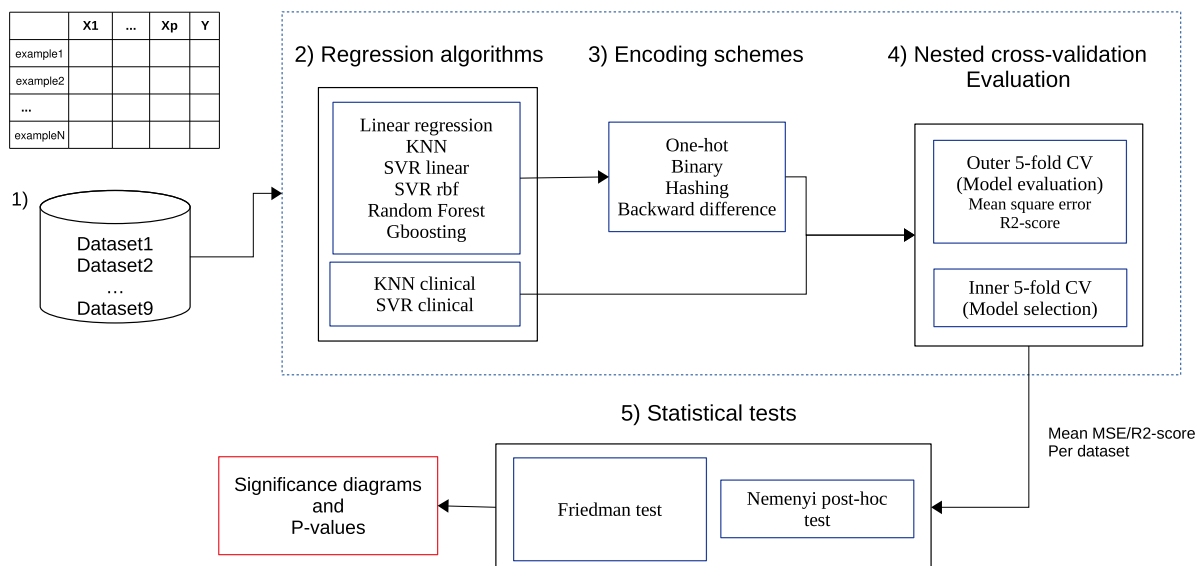


FIGURE 1. General overview of our approach. 1) having multiple tabular-like datasets with mixed variables, 2) we train different ML algorithms for regression 3) with different categorical encoding schemes (if necessary). 4) We evaluate the performance of all methods across multiple datasets and apply 5) statistical tests on the performance rankings in order to assess the overall performance of ML methods and encoding schemes with mixed-type variables.

or format, operating with them require special treatment to be transformed in numbers, understandable for algorithms. Many strategies to handle *mixed-type* data have been proposed in the literature, including encoding schemes for categorical features such as 'dummy' encoding or specialized metrics for clustering [10], [11]. However, it is unclear from the literature which encoding schemes are more appropriated for a certain class of regression algorithms, i.e., linear, tree-based, kernel-based models for regression. Conversely, deciding about which algorithm is the most appropriated for a regression task fixing an encoding strategy is critical, especially when *multiple* datasets are considered. Hence, in this paper, we propose a comparison framework to systematically evaluate and compare regression algorithms and encoding schemes to handle mixed-type variables datasets. Figure 1 shows a general overview of our approach.

Our methodology is presented in a real-life case study with engineering data from the tire industry. Typically tire-related data is highly dimensional, non-linear, and contains mixed variables, i.e., numerical and categorical features. For instance, automobile tires have molded into their sidewall the ISO code providing a generic description of the tire, see Figure 2. This code specifies the dimensions of the tire, i.e., tire width, aspect ratio, wheel diameter, and limitations such as the load-bearing ability, and maximum speed. A tire is built with different rubber types and composites describing their physical composition. Many of these variables are numerical, and others, such as speed rating, carcass type or belt type are categorical. Thus, handling adequately such mixture data is critical for the accuracy of ML algorithms in tire-performance prediction task.

Typically tire-engineers rely on physical or mechanical models to understand the relationship between variables and outcomes [12]. Because understanding the modeling process

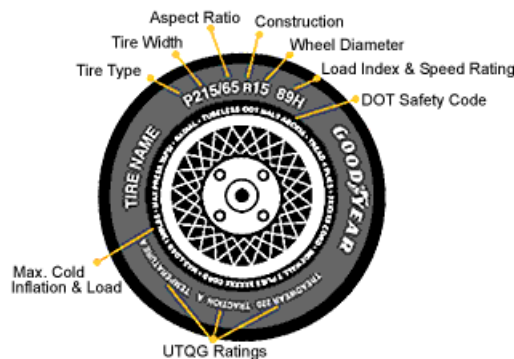


FIGURE 2. Sidewall features of a tire P215/65 R15 89H.

is critical in tire designing, we experimented with simple shallow models that may provide clearer explanations about the underlying predictions. The regression techniques we employed in this work are grouped in four categories, 1) *linear methods*., Linear regression (LR), 2) *kernel-based methods*.: support vector regression with linear and Gaussian kernels (SVR linear, SVR rbf), and an especial kernel capable of dealing with mixed-type variables, the clinical kernel (SVR clinical). 3) We use *non-parametric* K-nearest neighbors (KNN), and kernelized KNN (KNN clinical), and 4) *ensemble methods* with Random Forest (RF regressor) and Gradient boosting (GBoosting). We are particularly interested in the effect of encoding schemes for handling categorical variables, such as one-hot (dummy), binary, hashing, and backward difference encodings, in each group of algorithms mentioned above.

All techniques are evaluated under a nested cross-validation scheme with hyperparameter optimization, using the mean square error (MSE) and the coefficient of determination (R2-score) as evaluation metrics to the regression task.

Subsequently, we perform statistical inference on the reported R2-scores to test the significance of the differences between 1) regression algorithms for each encoding strategy and 2) encoding schemes for a given class of algorithms; across multiple datasets. We follow [13] to assess the differences between ML algorithms together with encoding schemes applying the non-parametric Friedman test and the Nemenyi post-hoc test. The effectiveness of this approach has been demonstrated in applications in science [14] and engineering [15] when comparing ML algorithms in classification tasks. Finally, the results are visualized with the significance diagrams [13] and p -value tables for pairwise comparisons.

It is worth mentioning that our purpose is not to perform a thorough investigation of ML methods with numerous encoding schemes. We rather present our contribution as a methodological approach providing recommendations for handling efficiently mixed-type data, i.e., categorical and numerical variables, while achieving high performances on regression tasks associated with multiple datasets. Because of the nature of the tire data we used, we consider only categorical variables ignoring any ordering between levels. However, the inclusion of ordinal variables is straightforward. Finally, although the conclusions are drawn in terms of tire-data, we emphasize that our approach can naturally be applied to any engineering datasets.

The paper is organized as follows. First, we review some related work, following by the presentation of the case study in section III. In section IV we introduce the ML algorithms, encoding schemes, statistical tests as well as the evaluation methodology. The experiments and results are presented in section V and the final conclusions in section VII.

II. RELATED WORK

A large number of ML methods for regression problems have been proposed and used in diverse engineering applications [6]–[8], [16], [17] to name some of them. Nevertheless, we focus on existing work that compares different approaches to regression problems for engineering applications. For instance, [18] compare four popular regression algorithms to establish the relationship between tire tread composites and filler system. More recently in [16], authors leverage numerical tire-size features to predict force and moment performances using different regression algorithms. Conversely, authors in [19] make a comparative study of categorical variable encodings for predicting vehicle properties using neural networks. A similarity measure to handle categorical variables is introduced in [20] and validated in KNN regression problems over twelve datasets. In [21] authors propose a hybrid decision tree algorithm for mixed categorical and numerical regression analysis. Their method is compared against five popular regression algorithms and, similar to us, they perform a statistical analysis of the performances of such methods over multiple datasets, but they used a single dummy variable encoding. More recent deep learning approaches for regression problems have proven to be very efficient in diverse application such as industrial

TABLE 1. Statistics of the considered datasets. Number of numerical, categorical and performance variables per dataset.

Dataset	# numerical	# categorical	# performances	# examples
D1	4	2	1	4694
D2	20	9	2	6975
D3	7	1	3	20000
D4	8	8	7	5633
D5	11	3	1	1159
D6	8	1	1	13000
D7	10	3	1	4694
D8	10	4	1	10158
D9	12	3	1	7959

surface defect detection [22], sustainable smart manufacturing in industry 4.0 [23] and short-long term load electricity forecasting [24]. Unlike the mentioned approaches, we present a rigorous statistical-based framework to compare and recommend regression algorithms considering many strategies to handle appropriately mixed variables over multiple datasets.

III. A CASE STUDY: TIRE-PERFORMANCES PREDICTOR

A. TIRE DATASETS

Our proposed methodology is presented in a case study from the tire industry. For this study, we collected several tire-related data from the automobile industry, containing tire-size features Figure 2, with performances tested at different conditions. Table 1 shows some statistics of the datasets we used in our study. Each dataset (rows) contains tire measurements for specific engineering target performances. Columns show the number of numerical and categorical features per dataset, and the associated performances to predict. For a complete description about tire tests in the automotive industry we refer the reader to [25].

In the following, we provide a brief description of the datasets.

- **D1:** This dataset contains measurements of the stretching force of a tire bead with different rim diameters, also known as bead compression test.
- **D2:** Consists of several measurements related to the rolling resistance force. This is a fundamental force acting in opposition to the motion when the tire rolls on a surface. Here we estimated two performances, denoted as D2_p0 and D2_p1
- **D3:** Was used before in [16], and contains relevant force and moments (F&M) measurements. Tire F&M are fundamental to characterize tire performance characteristics that highly influence the dynamics of the vehicle. That is, force and moment characteristics have to be designed such that the vehicle can easily be kept under driver control under diverse driving conditions. Three tire performances are estimated on this dataset, denoted as D3_p0, D3_p1 and D3_p2.
- **D4:** Consists of data related to plunger test, where a steel plunger is forced perpendicular to the tread of a mounted tire until the tire ruptures, or the plunger is stopped by reaching the rim. Here, seven performances are estimated, denoted as D4_p0, D4_p1, D4_p2, D4_p3, D4_p4, D4_p5 and D4_p6.

- **D5:** This dataset has measurements of the high-speed test, where the tire is exposed to different maximum speed scenarios where the tire can sustain.
- **D6:** Contains information of the contact patch of the tire which is touching the road surface. It is within this interaction area that tire forces and moments arise and that wear occurs.
- **D7:** This dataset contains measurements of the standard bead unseat test to determine the conditions to which a tire will stay on the rim.
- **D8:** Contains regulatory measurements related to noise perturbations of the tire in diverse speed conditions and surfaces of contact.
- **D9:** Contain diverse measurements of the tire wet grip associated with the ability of a tire to adhere to the road in wet conditions.

B. DATA PREPROCESSING

Once the datasets were collected by tire-engineers, data cleaning and preprocessing were carried out to remove inconsistencies and improve the quality of the data. To do so, we applied a series of transformations necessary to make the data modelling more efficient [26]. First, tire features were handcrafted and selected beforehand by tire-engineer experts, respecting physical constraints between variables and tire-performances. Second, anomalous observations were removed [27] by re-scaling and centering the data around zero and looking at points whose distance to the origin was larger than three times the standard deviation of the data, i.e., the z-score. Third, all datasets were standardized by removing its mean and scaling with its standard deviation. For certain kernel matrices such as the clinical kernel, we centered the matrix by normalizing to have zero mean.

C. TASK DESCRIPTION

We focus on building regression models on tire-related measurements to predict tire-performances across the datasets mentioned in Table 1. Also, the datasets include a mix of numerical and categorical variables. We use a well known family of regression algorithms from non-parametric, ensemble methods, and kernel-based algorithms. For the methods that do not handle directly categorical variables, we preprocess the datasets with different encoding schemes so that the considered algorithms can do computations on such data. Subsequently, we perform a rigorous statistical analysis on the performances of the methods across datasets, to state the significance of the machine learning methods and the encoding mechanism in tire-performance prediction.

IV. METHODS

In this section, we present our approach with the algorithms, the encoding schemes, and the statistical approaches we use in our analysis.

A. REGRESSION ALGORITHMS

The proposed methodology is applied to regression problems for the prediction of tire-performances. We have considered a wide range of linear and non-linear regression models

grouped in categories according to its nature: parametric (linear regression), non-parametric (K nearest neighbors), ensemble methods (Random Forest, Gradient Boosting), and kernel-based methods (support vector regression). A brief description of the techniques is introduced below.

- 1) **Linear regression** [28]: LR fits a linear model with real-valued coefficients to minimize the sum of squares of residuals between the observed targets in the data, and the predicted values by the linear approximation.
- 2) **K-nearest neighbors** [29]: This non-parametric algorithm is one of the simplest algorithms in ML. It uses features similarity to predict the target variable of new points. This means that a new point is assigned a value based on the average value of its K closest points in the training set. The Euclidean distance is used by default but can be extended to more general distances.
- 3) **Support vector regression** [30]: SVR algorithm is a variant of the popular support vector machines for classification. In its basic form SVR aims to fit a hyper-plane subject to all residuals having a value less than a non-negative coefficient ϵ determining the fitting accuracy. This model can be extended to a non-linear formulation whose basic principle is to map non-linear data to a higher dimensional space where the problem becomes linear. This transformation is done through a *kernel function* that has the property of being a dot product of feature mappings from the input to a Euclidean space. The problem is modeled as a quadratic optimization problem, and its solution provides real coefficients characterizing the so-called support vectors, i.e., training examples with associated null coefficients. A typical non-linear kernel is the radial basis function (Rbf) [30]:

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right) \quad (1)$$

for any $x, y \in R^p$ and σ a scaling parameter.

- 4) **The clinical kernel** [31]: This kernel has been proposed in the context of analyzing clinical data for patients. The main advantage is that it allows modelling numerical and categorical variables in a compact formulation without using any categorical encoding scheme. That is, for two data points $x, y \in R^p$ with p features, the clinical kernel [31] is defined as:

$$CK(x, y) = \frac{1}{p} \sum_{f=1}^p K_f(x, y) \quad (2)$$

where K_f is the kernel matrix of the feature f . Furthermore:

$$K_f(x, y) = \begin{cases} \frac{(\max_f - \min_f) - |x_f - y_f|}{\max_f - \min_f}, & \text{for } f \text{ continuous} \\ \delta(x_f = y_f), & \text{for } f \text{ categorical} \end{cases} \quad (3)$$

where $\delta(z) = 1$ when z is true and 0 otherwise, and x_f and y_f are the f^{th} feature of x and y . The \max_f and \min_f

represents their respective maximum and minimum value.

The choice of the appropriated kernel function may affect the performance of the ML algorithms in regression [32]. For this reason, we experimented with different linear and non-linear kernels within the SVR formulation, in particular the linear and Rbf kernels with different encoding schemes. In addition, the SVR clinical and KNN clinical are formulations of KNN and SVR when using the clinical kernel to handle mixed-variables in the regression tasks.

- 5) **Random Forest** [33]: RF is a popular algorithm based on aggregation principles for classification and regression problems. RF operates by constructing and training an ensemble of decision trees over a subset of randomly selected features. The decision trees outputs are combined to estimate a target value using any aggregation mechanism such as averaging individual responses.
- 6) **Gradient Boosting** [34]: GBoosting is an ensemble technique for classification and regression that aggregates the estimation of individual models (typically decision trees) improving the prediction from inaccurate inter-media estimations in a sequential manner.

B. ENCODING SCHEMES

Using categorical variables in regression problems is not a trivial task as there is not an explicit notion of ordering or semantic between its values or with the response variable. However, there are two main approaches used in the literature to deal with this difficulty. The most common is transforming categories into numerical values applying encoding schemes keeping semantic between categories' levels. The second one is designing specific algorithms for regression that handle internally mixed data, i.e., clinical kernel. In this work we will cover both alternatives focusing on the following encoding schemes:

- 1) **One-hot (dummy) encoding** [35]: This is one of the most popular encoding methods. For a categorical variable of cardinality C It creates $C - 1$ new binary features, with a value of 1 for the actual value and zero otherwise. It works well on linear models but is not suitable for variables with large C .
- 2) **Binary encoding** [35]: Similar to one-hot, but category levels are treated as positive integers and subsequently converted to binary digits. Each binary digit gets one column. For a variable of cardinality C it will add p new binary digits such that $p = \min\{i : C \leq 2^i, i \geq 0\}$ and $C = p + 1$.
- 3) **Hashing encoding** [36]: This method utilizes hash functions to map the levels of a categorical variable to numbers, which are themselves encoded to binary strings of a given dimension. As the levels are not memorized, it can deal with new levels gracefully and therefore scale to categories with large cardinality.

- 4) **Backward difference**: This is a kind of the so-called contrast encoding methods [37]. For a categorical variable with C levels, it creates new $C - 1$ variables of inter-level differences. That is, for each level, it calculates the difference between the mean output given a category, and the overall expected value of the dependent variable. It can be used with ordinal and categorical values.

C. MODEL COMPARISON AND STATISTICAL TESTS

Beyond the performances achieved for different algorithms, we use hypothesis testing techniques to provide statistical support to our analysis. Concretely we use non-parametric methods given that the strong assumptions required by parametric methods may not be satisfied in our datasets. Here we use the Friedman-Nemenyi test for comparing multiple algorithms across datasets [13]. The Friedman test provides evidence that the outcome of different regression models is statistically different. The Nemenyi post-hoc test assesses significant differences between individual algorithms.

The Friedman test is a non-parametric test based on the average ranked performances (R_j) of the regression models on each dataset. The Friedman statistics is calculated as

$$Q = \frac{12D}{K(K+1)} \sum_{j=1}^K \left(R_j - \frac{K+1}{2} \right)^2 \quad (4)$$

where D denotes the number of datasets, K the number of regression algorithms, and $R_j = \frac{1}{D} \sum_{i=1}^D r_i^j$ as the average rank of the algorithm j , where r_i^j denotes the rank of the j -th algorithm in the i -th dataset. Under the null hypothesis that all algorithms perform equally, the Q statistics is approximately distributed as a Chi-square χ_{K-1}^2 distribution with $K - 1$ degree of freedom. Therefore we can reject the null hypothesis and conclude that some algorithms perform better than other when Q is large enough, with the probability that $\chi_{K-1}^2 \geq Q$ as p -value.

If the null-hypothesis is rejected, we can proceed with a post-hoc test. The post-hoc Nemenyi test [13] is applied to report any significant difference between individual algorithms. This test states that the performance of various algorithms is significantly different if their average rank differs by at least the critical difference (CD):

$$CD = q_{\alpha, K} \sqrt{\frac{K(K+1)}{12D}} \quad (5)$$

where the critical values $q_{\alpha, K}$ are based on the Studentized range statistic divided by $\sqrt{2}$. Finally, the results of the Friedman-Nemenyi test can be visualized with the diagrams proposed by Demsar [13]. These diagrams show the mean ranked performances of the algorithms along with the critical difference such that the lower the ranking, the better the method. Horizontal lines connect algorithms that are not significantly different.

TABLE 2. R2-score of regression task with one-hot encoding.

Dataset	LR	SVR linear	SVR rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.23 ± 0.09	0.19 ± 0.10	0.26 ± 0.09	0.24 ± 0.09	0.31 ± 0.06	0.38 ± 0.04	0.17 ± 0.14	0.20 ± 0.07
D2_p0	0.85 ± 0.02	0.83 ± 0.04	0.94 ± 0.01	0.95 ± 0.02	0.92 ± 0.01	0.99 ± 0.01	0.86 ± 0.02	0.91 ± 0.02
D2_p1	0.85 ± 0.02	0.83 ± 0.04	0.94 ± 0.01	0.95 ± 0.02	0.92 ± 0.01	0.99 ± 0.01	0.86 ± 0.02	0.91 ± 0.02
D3_p0	0.78 ± 0.01	0.76 ± 0.01	0.96 ± 0.00	0.88 ± 0.01	0.95 ± 0.00	0.97 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
D3_p1	0.94 ± 0.00	0.94 ± 0.00	0.97 ± 0.00	0.95 ± 0.00	0.97 ± 0.00	0.98 ± 0.00	0.95 ± 0.00	0.95 ± 0.00
D3_p2	0.77 ± 0.01	0.76 ± 0.01	0.84 ± 0.00	0.81 ± 0.01	0.82 ± 0.01	0.87 ± 0.01	0.84 ± 0.01	0.85 ± 0.01
D4_p0	0.94 ± 0.02	0.93 ± 0.04	0.96 ± 0.01	0.96 ± 0.02	0.98 ± 0.00	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.01
D4_p1	0.94 ± 0.02	0.93 ± 0.04	0.95 ± 0.01	0.96 ± 0.03	0.97 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.97 ± 0.01
D4_p2	0.94 ± 0.02	0.93 ± 0.04	0.95 ± 0.01	0.96 ± 0.02	0.97 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.97 ± 0.01
D4_p3	0.96 ± 0.02	0.94 ± 0.04	0.96 ± 0.02	0.98 ± 0.02	0.99 ± 0.01	0.99 ± 0.01	0.97 ± 0.03	0.98 ± 0.01
D4_p4	0.94 ± 0.02	0.92 ± 0.04	0.95 ± 0.01	0.96 ± 0.03	0.97 ± 0.01	0.97 ± 0.01	0.95 ± 0.02	0.97 ± 0.01
D4_p5	0.94 ± 0.02	0.93 ± 0.03	0.95 ± 0.01	0.96 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.01
D4_p6	0.90 ± 0.03	0.88 ± 0.04	0.92 ± 0.03	0.91 ± 0.02	0.93 ± 0.02	0.94 ± 0.01	0.91 ± 0.03	0.92 ± 0.01
D5	0.96 ± 0.01	0.95 ± 0.02	0.95 ± 0.01	0.93 ± 0.03	0.92 ± 0.02	0.94 ± 0.02	0.89 ± 0.06	0.86 ± 0.08
D6	0.22 ± 0.01	0.20 ± 0.01	0.37 ± 0.02	0.34 ± 0.01	0.42 ± 0.02	0.53 ± 0.02	0.46 ± 0.02	0.45 ± 0.02
D7	0.76 ± 0.08	0.76 ± 0.08	0.82 ± 0.07	0.80 ± 0.07	0.81 ± 0.07	0.80 ± 0.08	0.75 ± 0.10	0.77 ± 0.08
D8	0.38 ± 0.12	0.39 ± 0.11	0.39 ± 0.12	0.40 ± 0.11	0.41 ± 0.12	0.21 ± 0.18	0.33 ± 0.12	0.33 ± 0.12
D9	0.55 ± 0.14	0.52 ± 0.19	0.57 ± 0.10	0.61 ± 0.09	0.63 ± 0.08	0.60 ± 0.11	0.36 ± 0.21	0.49 ± 0.14

D. EVALUATION METHODOLOGY

We evaluate the performance of the regression techniques using nested cross-validation (CV) approach. This method is commonly used to reduce the bias in the generalization error induced by the random splitting of the dataset. The inner CV is used to select the optimal model and the outer CV to estimate the generalization error of the method. In the outer 5-fold CV loop, 80% of the data (four folds) is provided as a training set to the inner 5-fold CV, and the remaining 20%, i.e., the hold-out fold, is used as testing set to evaluate the performance of the model. Within the inner 5-fold CV, we tune the parameters of the ML models. i.e., The ϵ , σ and C for SVR, number of trees for Random Forest and GBoosting, and k for KNN, selecting the best model using the inner hold-out fold as validation set within a grid-search hyperparameter optimization scheme [38], [39]. That is, grid-search exhaustively generates a combination of parameters from a grid of predefined values in order to train ML algorithms and select the best model. Finally, we report the average and standard deviation of the mean square error (MSE) and the coefficient of determination (R2-score) of the outer CV folds.

V. EXPERIMENTS AND RESULTS

In this section, we present our comparison methodology and results. That is, following the scheme of Figure 1, for each dataset of Table 1, we transform the categorical variables applying the encoding schemes of section IV-B. Subsequently, we perform regression tasks applying the algorithms introduced previously in section IV-A, following the evaluation setting of section IV-D. For the algorithms that can handle mixed variables, we perform regression without any encoding. Finally, we provide the overall evaluation of our models, applying the statistical tests presented in IV-C. Such tests were performed on the R2-scores only, but similar conclusions are achieved when applied on mse. In this paper, the Friedman test is evaluated with a significance level of 0.01 followed by the post-hoc Nemenyi test with a significance of 0.05 [13]. Our approach was coded in Python 3.7 with sklearn 0.22.2, and run on a standard Laptop Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz with 16Gb of RAM.

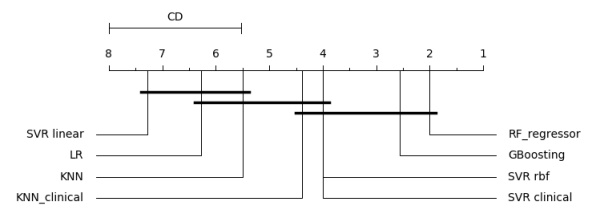


FIGURE 3. Results of Nemenyi test for one-hot encoded algorithms. Groups of algorithms that are not significantly different at $p = 0.05$ are connected.

A. BEST ALGORITHM FOR ENCODING STRATEGY

The first part of our experimental setting aims to answer the question: For a given encoding scheme, which machine learning algorithm is the most appropriated to predict tire-performances?. As the strategy to manage mixed-variables is critical, we complete our analysis for each encoding scheme and summarize the overall outcome in the discussion section.

1) RESULTS WITH ONE-HOT ENCODING

Table 2 shows the Mean and standard deviation of the R2-score for the regression tasks.

The critical value of χ^2_7 is 18.47, which is lower than Friedman's statistic $Q = 67.259$. As a consequence, we reject the null hypothesis that all algorithm's performance is equivalent. Regarding this rejection, the post-hoc Nemenyi test is applied. The significance diagram in Fig. 3 shows the average performance ranks of the algorithms along with the Nemenyi's critical difference tail ($CD = 2.474$).

From the figures 3, 4, we can see that:

- GBoosting and Random Forest are significantly better than KNN, LR and SVR linear.
- The SVR clinical and SVR Rbf kernels are equivalent and significantly better than SVR linear.
- The SVR linear is significantly worse than KNN clinical, SVR clinical, SVR Rbf, Random Forest and GBoosting.

2) RESULTS WITH BINARY ENCODING

Table 3 shows the Mean and standard deviation of the R2-score for the regression task across datasets. The Friedman statistics is $Q = 71.093$, which is greater than the critical

TABLE 3. R2-score of regression task with Binary encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.23 ± 0.11	0.20 ± 0.12	0.28 ± 0.11	0.26 ± 0.10	0.33 ± 0.07	0.37 ± 0.08	0.09 ± 0.15	0.15 ± 0.24
D2_p0	0.74 ± 0.10	0.72 ± 0.11	0.93 ± 0.03	0.95 ± 0.02	0.88 ± 0.04	0.98 ± 0.01	0.58 ± 0.29	0.68 ± 0.22
D2_p1	0.74 ± 0.10	0.72 ± 0.11	0.93 ± 0.03	0.95 ± 0.02	0.87 ± 0.04	0.98 ± 0.01	0.58 ± 0.29	0.68 ± 0.22
D3_p0	0.78 ± 0.01	0.76 ± 0.01	0.96 ± 0.00	0.88 ± 0.01	0.95 ± 0.00	0.97 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
D3_p1	0.94 ± 0.00	0.94 ± 0.00	0.98 ± 0.00	0.95 ± 0.00	0.97 ± 0.00	0.98 ± 0.00	0.95 ± 0.00	0.95 ± 0.00
D3_p2	0.76 ± 0.01	0.76 ± 0.01	0.84 ± 0.00	0.81 ± 0.01	0.82 ± 0.01	0.87 ± 0.01	0.84 ± 0.01	0.85 ± 0.01
D4_p0	0.86 ± 0.08	0.78 ± 0.16	0.96 ± 0.02	0.93 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p1	0.86 ± 0.08	0.78 ± 0.16	0.96 ± 0.02	0.94 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p2	0.86 ± 0.07	0.79 ± 0.16	0.96 ± 0.02	0.94 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p3	0.86 ± 0.08	0.79 ± 0.16	0.96 ± 0.02	0.95 ± 0.03	0.98 ± 0.02	0.98 ± 0.01	0.97 ± 0.02	0.97 ± 0.02
D4_p4	0.86 ± 0.07	0.79 ± 0.15	0.96 ± 0.02	0.94 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.96 ± 0.02
D4_p5	0.86 ± 0.07	0.79 ± 0.15	0.96 ± 0.02	0.94 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p6	0.81 ± 0.09	0.78 ± 0.14	0.93 ± 0.03	0.90 ± 0.03	0.93 ± 0.02	0.94 ± 0.02	0.92 ± 0.03	0.93 ± 0.02
D5	0.96 ± 0.01	0.95 ± 0.02	0.95 ± 0.01	0.93 ± 0.03	0.92 ± 0.02	0.94 ± 0.02	0.89 ± 0.06	0.86 ± 0.08
D6	0.22 ± 0.02	0.21 ± 0.02	0.35 ± 0.02	0.33 ± 0.01	0.38 ± 0.02	0.51 ± 0.02	0.43 ± 0.02	0.42 ± 0.02
D7	0.78 ± 0.06	0.77 ± 0.06	0.83 ± 0.06	0.82 ± 0.06	0.83 ± 0.06	0.83 ± 0.07	0.78 ± 0.08	0.80 ± 0.06
D8	0.36 ± 0.15	0.39 ± 0.14	0.40 ± 0.12	0.41 ± 0.11	0.40 ± 0.12	0.19 ± 0.19	0.32 ± 0.12	0.33 ± 0.11
D9	0.58 ± 0.10	0.56 ± 0.15	0.60 ± 0.10	0.61 ± 0.09	0.63 ± 0.09	0.60 ± 0.11	0.46 ± 0.15	0.49 ± 0.14

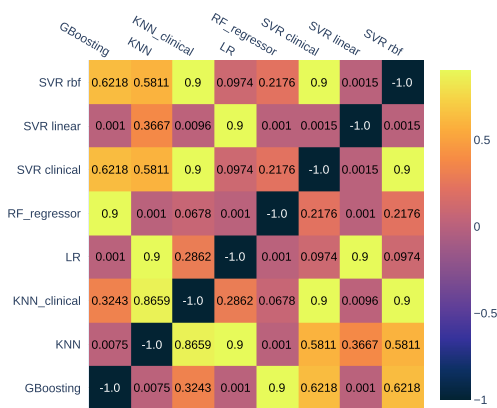


FIGURE 4. The p-values for the one-hot encoding obtained with the post-hoc pairwise multiple comparison according to Nemenyi's test.

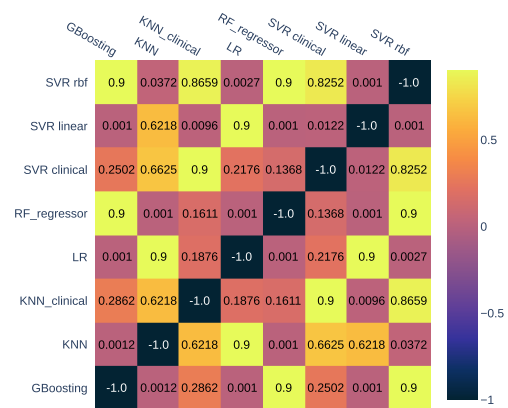


FIGURE 6. The p-values for the binary encoding obtained with the post-hoc pairwise multiple comparison according to Nemenyi's test.

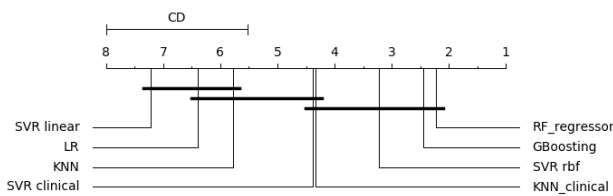


FIGURE 5. Results of Nemenyi test for binary encoded algorithms. Groups of algorithms that are not significantly different at $p = 0.05$ are connected.

value $\chi_7^2 = 18.47$. Thus, the null hypothesis is rejected with a significance of 0.01, and we proceed with the Nemenyi test. From the significance diagram of Figure 5 and 6 we can state that:

- GBoosting, Random Forest and SVR Rbf are significantly better than KNN, LR and SVR linear.
- The SVR clinical significantly better than SVR linear
- The SVR linear is significantly worse than KNN clinical, SVR clinical, SVR Rbf, Random Forest and GBoosting.

3) RESULTS WITH HASHING ENCODING

Table 4 shows the Mean and standard deviation of the R2-score for the regression task across datasets. The critical value of χ_7^2 is 18.47, which is lower than Friedman's statistic $Q = 70.074$. We reject the null hypothesis and conclude that

the algorithm's performance is not equivalent. The critical difference, according to the Nemenyi test, is $CD = 2.474$. Its significance diagram is shown in Fig. 7 showing the following results:

- Random Forest is significantly better than KNN, LR, SVR clinical, SVR linear.
- The SVR Rbf is significantly better than LR and SVR linear
- The SVR linear is significantly worse than KNN clinical, SVR clinical, SVR Rbf, Random Forest and GBoosting.

4) RESULTS WITH BACKWARD DIFFERENCE ENCODING

Table 5 shows the Mean and standard deviation of the R2-score for the regression task across datasets.

Here, the Friedman statistics $Q = 70.033$ is greater than the critical value $\chi_7^2 = 18.47$. Thus, we reject the null hypothesis with a significance of 0.01, and we conclude that the algorithms perform differently. The critical difference for the Nemenyi test is $CD = 2.4747$. Thus, regarding the significance diagram of Figure 9, we can conclude that:

- GBoosting and Random Forest are significantly better than KNN, LR and SVR linear
- The SVM Rbf is significantly better than KNN and SVR linear

TABLE 4. R2-score of regression task with hashing encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.23 ± 0.11	0.20 ± 0.12	0.28 ± 0.11	0.26 ± 0.10	0.33 ± 0.07	0.37 ± 0.08	0.23 ± 0.09	0.15 ± 0.24
D2_p0	0.75 ± 0.11	0.73 ± 0.09	0.96 ± 0.03	0.95 ± 0.02	0.88 ± 0.03	0.98 ± 0.01	0.60 ± 0.29	0.68 ± 0.22
D2_p1	0.75 ± 0.11	0.73 ± 0.09	0.96 ± 0.03	0.95 ± 0.02	0.88 ± 0.03	0.98 ± 0.01	0.60 ± 0.29	0.68 ± 0.22
D3_p0	0.78 ± 0.01	0.76 ± 0.01	0.96 ± 0.00	0.88 ± 0.01	0.95 ± 0.00	0.97 ± 0.00	0.94 ± 0.00	0.93 ± 0.00
D3_p1	0.94 ± 0.00	0.94 ± 0.00	0.97 ± 0.00	0.95 ± 0.00	0.97 ± 0.00	0.98 ± 0.00	0.95 ± 0.00	0.95 ± 0.00
D3_p2	0.77 ± 0.01	0.76 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.82 ± 0.01	0.87 ± 0.01	0.84 ± 0.01	0.85 ± 0.01
D4_p0	0.86 ± 0.08	0.78 ± 0.17	0.97 ± 0.01	0.93 ± 0.03	0.97 ± 0.01	0.97 ± 0.02	0.96 ± 0.02	0.97 ± 0.02
D4_p1	0.86 ± 0.08	0.78 ± 0.16	0.97 ± 0.01	0.94 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p2	0.86 ± 0.07	0.78 ± 0.15	0.97 ± 0.01	0.94 ± 0.03	0.97 ± 0.02	0.97 ± 0.01	0.96 ± 0.03	0.97 ± 0.02
D4_p3	0.86 ± 0.09	0.78 ± 0.17	0.97 ± 0.01	0.95 ± 0.03	0.98 ± 0.01	0.98 ± 0.01	0.97 ± 0.02	0.97 ± 0.02
D4_p4	0.86 ± 0.07	0.79 ± 0.15	0.97 ± 0.01	0.94 ± 0.02	0.97 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.96 ± 0.02
D4_p5	0.86 ± 0.07	0.78 ± 0.15	0.97 ± 0.01	0.94 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.97 ± 0.02
D4_p6	0.81 ± 0.09	0.78 ± 0.14	0.93 ± 0.03	0.90 ± 0.03	0.93 ± 0.02	0.94 ± 0.02	0.92 ± 0.03	0.93 ± 0.02
D5	0.96 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.93 ± 0.03	0.92 ± 0.02	0.94 ± 0.02	0.89 ± 0.05	0.86 ± 0.08
D6	0.22 ± 0.02	0.21 ± 0.02	0.34 ± 0.02	0.33 ± 0.01	0.39 ± 0.02	0.51 ± 0.02	0.43 ± 0.02	0.42 ± 0.02
D7	0.78 ± 0.06	0.77 ± 0.07	0.83 ± 0.06	0.82 ± 0.06	0.83 ± 0.06	0.83 ± 0.06	0.78 ± 0.08	0.80 ± 0.06
D8	0.29 ± 0.18	0.33 ± 0.15	0.39 ± 0.13	0.41 ± 0.11	0.39 ± 0.13	0.20 ± 0.19	0.28 ± 0.14	0.33 ± 0.11
D9	0.58 ± 0.11	0.55 ± 0.16	0.57 ± 0.09	0.61 ± 0.09	0.62 ± 0.10	0.59 ± 0.12	0.47 ± 0.13	0.49 ± 0.14

TABLE 5. R2-score of regression task with Backward difference encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.23 ± 0.11	0.20 ± 0.12	0.28 ± 0.11	0.26 ± 0.10	0.33 ± 0.08	0.37 ± 0.08	0.02 ± 0.15	0.15 ± 0.24
D2_p0	0.76 ± 0.10	0.75 ± 0.07	0.86 ± 0.13	0.95 ± 0.02	0.88 ± 0.04	0.98 ± 0.01	0.57 ± 0.26	0.68 ± 0.22
D2_p1	0.76 ± 0.10	0.75 ± 0.07	0.86 ± 0.13	0.95 ± 0.02	0.88 ± 0.04	0.98 ± 0.01	0.57 ± 0.26	0.68 ± 0.22
D3_p0	0.78 ± 0.01	0.76 ± 0.01	0.96 ± 0.00	0.88 ± 0.01	0.95 ± 0.00	0.97 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
D3_p1	0.94 ± 0.00	0.94 ± 0.00	0.97 ± 0.00	0.95 ± 0.00	0.97 ± 0.00	0.98 ± 0.00	0.95 ± 0.00	0.95 ± 0.00
D3_p2	0.77 ± 0.01	0.76 ± 0.01	0.84 ± 0.00	0.81 ± 0.01	0.82 ± 0.01	0.87 ± 0.01	0.85 ± 0.01	0.85 ± 0.01
D4_p0	0.94 ± 0.03	0.92 ± 0.04	0.95 ± 0.02	0.93 ± 0.03	0.97 ± 0.01	0.97 ± 0.02	0.94 ± 0.03	0.97 ± 0.02
D4_p1	0.93 ± 0.03	0.91 ± 0.05	0.95 ± 0.02	0.94 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.94 ± 0.03	0.97 ± 0.02
D4_p2	0.94 ± 0.03	0.92 ± 0.05	0.95 ± 0.02	0.94 ± 0.03	0.97 ± 0.01	0.98 ± 0.01	0.94 ± 0.03	0.97 ± 0.02
D4_p3	0.95 ± 0.03	0.93 ± 0.04	0.95 ± 0.02	0.95 ± 0.03	0.98 ± 0.01	0.98 ± 0.01	0.95 ± 0.03	0.97 ± 0.02
D4_p4	0.94 ± 0.03	0.92 ± 0.05	0.95 ± 0.02	0.94 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.94 ± 0.03	0.96 ± 0.02
D4_p5	0.94 ± 0.03	0.92 ± 0.05	0.95 ± 0.02	0.94 ± 0.02	0.97 ± 0.01	0.98 ± 0.01	0.94 ± 0.03	0.97 ± 0.02
D4_p6	0.90 ± 0.03	0.89 ± 0.04	0.92 ± 0.03	0.90 ± 0.03	0.93 ± 0.02	0.94 ± 0.02	0.90 ± 0.03	0.93 ± 0.02
D5	0.96 ± 0.01	0.95 ± 0.02	0.95 ± 0.01	0.93 ± 0.03	0.92 ± 0.02	0.94 ± 0.02	0.89 ± 0.06	0.86 ± 0.08
D6	0.22 ± 0.02	0.21 ± 0.02	0.36 ± 0.02	0.33 ± 0.01	0.39 ± 0.02	0.51 ± 0.02	0.43 ± 0.02	0.42 ± 0.02
D7	0.78 ± 0.06	0.77 ± 0.06	0.83 ± 0.06	0.82 ± 0.06	0.83 ± 0.06	0.83 ± 0.06	0.78 ± 0.08	0.80 ± 0.06
D8	0.37 ± 0.14	0.40 ± 0.12	0.39 ± 0.12	0.41 ± 0.11	0.41 ± 0.12	0.18 ± 0.20	0.32 ± 0.13	0.33 ± 0.11
D9	0.55 ± 0.13	0.52 ± 0.19	0.56 ± 0.10	0.61 ± 0.09	0.63 ± 0.09	0.61 ± 0.10	0.37 ± 0.21	0.49 ± 0.14

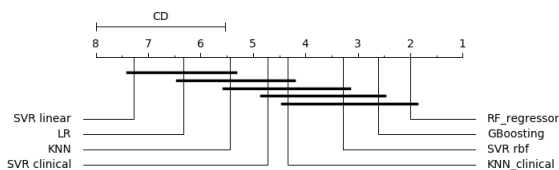


FIGURE 7. Results of Nemenyi test for hashing encoded algorithms. Groups of algorithms that are not significantly different at $p = 0.05$ are connected.

- The SVR linear is significantly worse than KNN clinical, SVR clinical, SVR Rbf, Random Forest and GBoosting

5) DISCUSSION OF ENCODING SCHEMES

In this experimental setting, we have shown the effect of different strategies for handling mixed variables in the precision of the regression algorithms. Overall, the experiments reveal that ensemble methods for regression, i.e., Random Forest and GBoosting, perform significantly better than non-parametric (KNN) and linear models, i.e., linear regression (LR) and SVR linear, in all encoding schemes. This can be explained because tire-related data is highly dimensional and non-linear, i.e., tire performances have complex physical dependencies with tire features and compounds, and thus non-linear ensemble models will outperform simpler linear models. Besides, the Random Forest model uses a collection of trees to make its predictions selecting a random subset

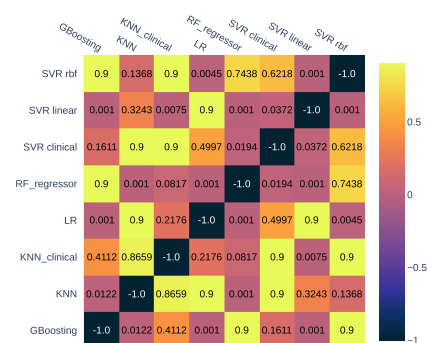


FIGURE 8. The p -values for the hashing encoding obtained with the post-hoc pairwise multiple comparison according to Nemenyi's test.

of input features, reducing the search space where each tree is optimizing. Hence, ensemble methods can better handle the curse of dimensionality introduced by different encoding schemes, therefore improving its generalization capability. On the other hand, there is not enough evidence to claim that ensemble methods are better than non-linear kernels, i.e., SVR Rbf and SVR clinical. Besides, it is clear that SVR linear performance is significantly the worst of all methods, no matter the encoding strategy used. Finally, SVR Rbf and SVR clinical are equivalent methods under one-hot encoding, and unlike ensemble methods, they provide a meaningful way to compare tires for further mining purposes. This will be discussed with more details in section VI.

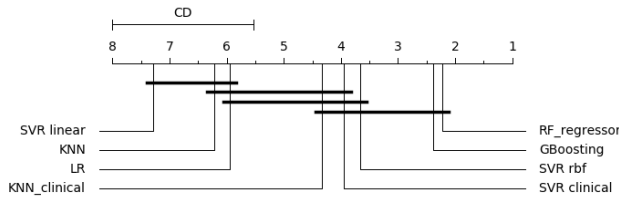


FIGURE 9. Results of Nemenyi test for backward difference encoded algorithms. Groups of algorithms that are not significantly different at $p = 0.05$ are connected.

B. BEST ENCODING FOR CLASSES OF ALGORITHMS

In the second part of our experiments, we address the problem of choosing the best encoding for classes (groups) of algorithms, in particular for the most performing methods from the previous section. We consider ensemble, kernel-based, and non-parametric models as groups under analysis.

1) ENSEMBLE METHODS

Here we investigate the best encoding scheme for Random Forest and GBoosting algorithms.¹ We apply the Friedman test independently on Random Forest and GBoosting performances from Tables 2,3,4,5 to test the hypothesis that the selected algorithm has equivalent performances across encoding strategies.

Starting with Random Forest, the critical value of χ^2_3 for a significance level of 0.01 is 11.34, which is greater than the Friedman’s statistic $Q = 5.267$, meaning that we fail to reject the null hypothesis. Similarly, Friedman’s statistic for GBoosting is $Q = 1.80$, so that we fail to reject the hypothesis that all encoding methods perform equally. As a consequence, we verify that *there is not a favorite* encoding strategy to perform regression with ensemble methods in this tire-performance data.

2) KERNEL-BASED REGRESSION

Here we investigate the effect of encoding schemes on SVR Rbf and SVR linear methods. As before, we perform the Friedman test independently on the performance from tables 2,3,4 and 5, to test the hypothesis that the performance of the given algorithm is equivalent across different encoding systems.

The critical value for the χ^2_3 with a significance of 0.01 is 11.34, which is greater than the Friedman’s statistics $Q = 2.799$. Thus, we fail to reject the null hypothesis that the SVR linear performances are equivalent across encodings. On the other hand, $Q = 20.847$ for SVR Rbf (including the SVR clinical), which is greater than the critical value χ^2_4 of 13.276 at 0.01 of significance. The null hypothesis is rejected and the Nemenyi critical difference $CD = 1.43$ is shown in the significance diagram, Figures 11 and 12

Here we observe that the considered encoding schemes are equivalent under SVR linear models, which is expected because this kernels can handle large dimensional data better

¹Although tree based methods are known for nativity handle categorical variables without any transformation, most of implementations require an encoding preprocessing step.

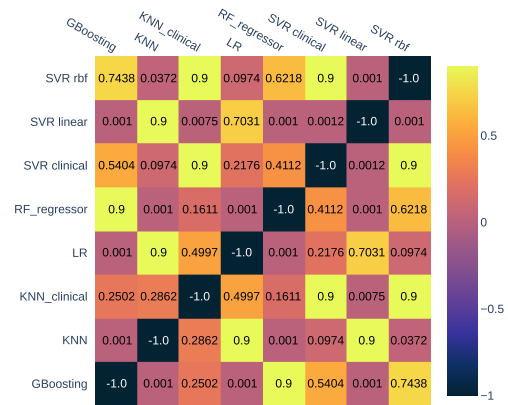


FIGURE 10. The p -values for the backward encoding obtained with the post-hoc pairwise multiple comparison according to Nemenyi’s test.

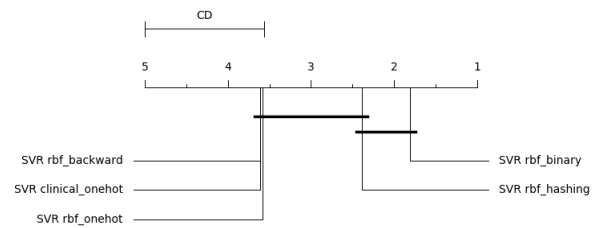


FIGURE 11. Results of Nemenyi test for SVR Rbf. Groups of encodings that are not significantly different at $p = 0.05$ are connected.

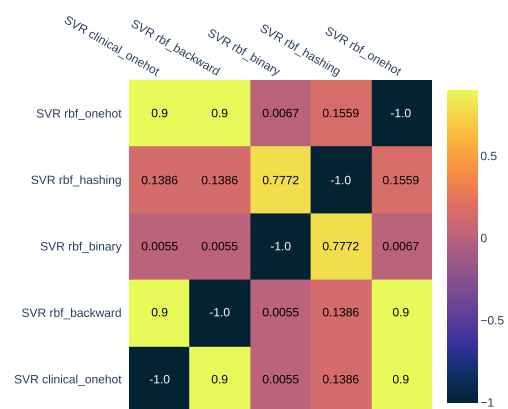


FIGURE 12. The p -values for the SVR Rbf regression obtained with the post-hoc pairwise multiple comparison according to Nemenyi’s test.

than other models, impart due to the excessive regularization that occurs [40]. On the other hand, the binary encoding is significantly better than backward difference, clinical, and one-hot encoding for the SVR Rbf. Indeed, the Gaussian kernel is more sensitive to the curse of dimensionality, performing better in encoding schemes with lower dimensions.

3) K-NEAREST NEIGHBOR REGRESSION

Here we investigate the effect of the considered encodings on KNN regression (including KNN clinical). As before, we perform the Friedman test on the performances taken from tables 2,3,4 and 5, to test the hypothesis that KNN performs similarly across different encoding strategies.

TABLE 6. MSE of regression task with one-hot encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.77 ± 0.11	0.81 ± 0.12	0.74 ± 0.11	0.76 ± 0.10	0.69 ± 0.08	0.62 ± 0.08	0.82 ± 0.15	0.79 ± 0.24
D2_p0	0.15 ± 0.10	0.17 ± 0.07	0.06 ± 0.13	0.05 ± 0.02	0.08 ± 0.04	0.01 ± 0.01	0.14 ± 0.26	0.09 ± 0.22
D2_p1	0.15 ± 0.10	0.17 ± 0.07	0.06 ± 0.13	0.05 ± 0.02	0.08 ± 0.04	0.01 ± 0.01	0.14 ± 0.26	0.09 ± 0.22
D3_p0	0.22 ± 0.01	0.24 ± 0.01	0.04 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.07 ± 0.00	0.07 ± 0.00
D3_p1	0.06 ± 0.00	0.06 ± 0.00	0.03 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.05 ± 0.00	0.05 ± 0.00
D3_p2	0.23 ± 0.01	0.23 ± 0.01	0.16 ± 0.00	0.19 ± 0.01	0.18 ± 0.01	0.13 ± 0.01	0.15 ± 0.01	0.15 ± 0.01
D4_p0	0.05 ± 0.03	0.06 ± 0.04	0.04 ± 0.02	0.03 ± 0.03	0.02 ± 0.01	0.02 ± 0.02	0.04 ± 0.03	0.03 ± 0.02
D4_p1	0.06 ± 0.03	0.07 ± 0.05	0.05 ± 0.02	0.04 ± 0.03	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.03	0.03 ± 0.02
D4_p2	0.06 ± 0.03	0.07 ± 0.05	0.05 ± 0.02	0.04 ± 0.03	0.03 ± 0.01	0.02 ± 0.01	0.04 ± 0.03	0.03 ± 0.02
D4_p3	0.04 ± 0.03	0.05 ± 0.04	0.03 ± 0.02	0.02 ± 0.03	0.01 ± 0.01	0.01 ± 0.01	0.03 ± 0.03	0.02 ± 0.02
D4_p4	0.06 ± 0.03	0.07 ± 0.05	0.05 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.02 ± 0.01	0.05 ± 0.03	0.03 ± 0.02
D4_p5	0.05 ± 0.03	0.07 ± 0.05	0.05 ± 0.02	0.04 ± 0.02	0.02 ± 0.01	0.02 ± 0.01	0.04 ± 0.03	0.03 ± 0.02
D4_p6	0.10 ± 0.03	0.12 ± 0.04	0.08 ± 0.03	0.09 ± 0.03	0.06 ± 0.02	0.06 ± 0.02	0.08 ± 0.03	0.07 ± 0.02
D5	0.04 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	0.06 ± 0.03	0.07 ± 0.02	0.06 ± 0.02	0.11 ± 0.06	0.14 ± 0.08
D6	0.78 ± 0.02	0.80 ± 0.02	0.63 ± 0.02	0.66 ± 0.01	0.58 ± 0.02	0.47 ± 0.02	0.54 ± 0.02	0.55 ± 0.02
D7	0.24 ± 0.06	0.24 ± 0.06	0.18 ± 0.06	0.19 ± 0.06	0.19 ± 0.06	0.19 ± 0.06	0.25 ± 0.08	0.23 ± 0.06
D8	0.56 ± 0.14	0.55 ± 0.12	0.55 ± 0.12	0.54 ± 0.11	0.53 ± 0.12	0.70 ± 0.20	0.60 ± 0.13	0.59 ± 0.11
D9	0.43 ± 0.13	0.45 ± 0.19	0.42 ± 0.10	0.37 ± 0.09	0.36 ± 0.09	0.38 ± 0.10	0.61 ± 0.21	0.49 ± 0.14

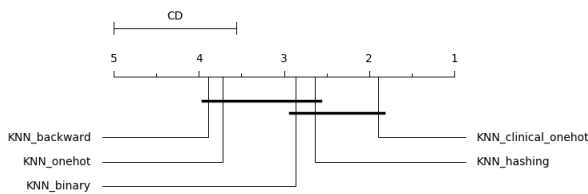


FIGURE 13. Results of Nemenyi test for KNN. Groups of encodings that are not significantly different at $p = 0.05$ are connected.

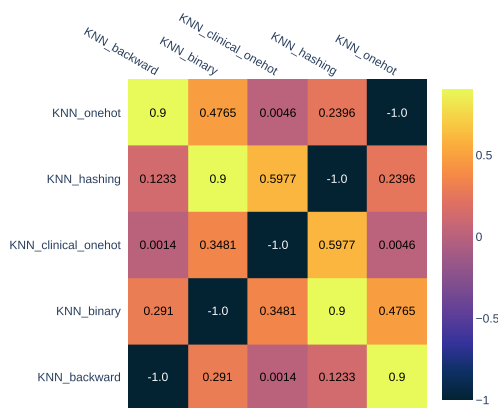


FIGURE 14. The p -values for KNN regression obtained with the post-hoc pairwise multiple comparison according to Nemenyi's test.

The critical value for the χ^2_4 with a significance of 0.01 is 13.276, which is lower than the Friedman's statistics $Q = 19.520$. Thus, we reject the null hypothesis, and after applying the Nemenyi post-hoc test, we achieve a critical difference of $CD = 1.43$ which is shown in the significance diagram, Figures 13 and 14

KNN is well known to be sensitive to high dimensions. Therefore, it is shown that KNN clinical is significantly better than backward difference and one-hot encoding.

VI. A TIRE-SIMILARITY FUNCTION

As a side benefit, the previous analysis allows us to define a similarity metric for tire-comparison. Similarity in data mining is an important concept for searching, comparing, and retrieving objects from a database. Designing a similarity measure for tire-based data is not a trivial task as the *mixed-type* nature of its variables and because there is not a general

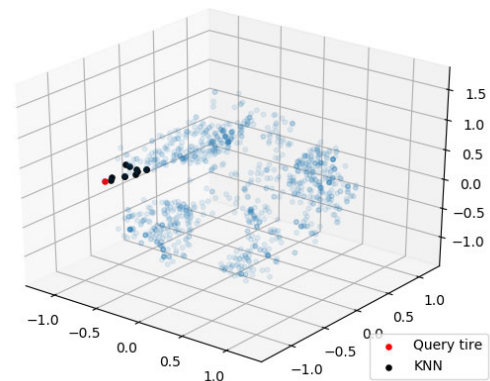


FIGURE 15. K-nearest neighbors with clinical kernel (MDS projection).

rule of what a 'good' similarity should be. Roughly speaking, a similarity function $f : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ is a real-valued function that takes two input tires x and y and assigns a high value when they are 'close' whereas assigns a lower value when x, y are far away in a Euclidean space. As we showed in section 3, this notion is implicit in the definition of kernels. Although the intuition of kernels as measures of similarity is not always obvious [41], there are cases where this notion coincides. In addition, we hypothesize that a good similarity function for tires should be good enough to predict tire performances. Thus, supported by our results from the previous section, we consider the Gaussian (Rbf) and the clinical kernel functions as relevant similarity measures for tires.

QUERYING A REFERENCE TIRE

We show the proposed tire-similarity qualitatively in a common industrial application of the query by example principle. That is, given a reference existing tire from the dataset, we retrieve the nearest tires according to the similarity function. As a proof of concept, we provide a visualization of the D3 from Table 1, i.e., the force and moments data, by embedding the tires in a three-dimensional space preserving the clinical-kernel similarity. The pairwise dissimilarity matrix is projected to 3d coordinates by means of the Multidimensional scaling (MDS) [42] algorithm so that the embedded tires preserve the original clinical-kernel distances.

TABLE 7. MSE of regression task with binary encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.77 ± 0.14	0.81 ± 0.15	0.73 ± 0.13	0.76 ± 0.14	0.69 ± 0.10	0.61 ± 0.09	0.79 ± 0.13	0.79 ± 0.05
D2_p0	0.17 ± 0.03	0.20 ± 0.04	0.03 ± 0.01	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.13 ± 0.02	0.09 ± 0.02
D2_p1	0.17 ± 0.03	0.20 ± 0.04	0.03 ± 0.01	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.13 ± 0.02	0.09 ± 0.02
D3_p0	0.22 ± 0.01	0.24 ± 0.02	0.04 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.07 ± 0.00	0.07 ± 0.00
D3_p1	0.06 ± 0.01	0.06 ± 0.01	0.02 ± 0.00	0.05 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.05 ± 0.00	0.05 ± 0.00
D3_p2	0.24 ± 0.01	0.24 ± 0.01	0.16 ± 0.01	0.19 ± 0.01	0.18 ± 0.01	0.13 ± 0.01	0.16 ± 0.01	0.15 ± 0.01
D4_p0	0.06 ± 0.02	0.06 ± 0.03	0.03 ± 0.01	0.03 ± 0.02	0.02 ± 0.01	0.02 ± 0.00	0.04 ± 0.00	0.03 ± 0.01
D4_p1	0.06 ± 0.02	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.04 ± 0.01	0.03 ± 0.01
D4_p2	0.06 ± 0.02	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.04 ± 0.01	0.03 ± 0.01
D4_p3	0.04 ± 0.02	0.04 ± 0.03	0.02 ± 0.01	0.02 ± 0.02	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01
D4_p4	0.06 ± 0.02	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.05 ± 0.01	0.03 ± 0.01
D4_p5	0.05 ± 0.02	0.05 ± 0.03	0.04 ± 0.01	0.04 ± 0.02	0.02 ± 0.01	0.03 ± 0.02	0.04 ± 0.01	0.03 ± 0.01
D4_p6	0.12 ± 0.02	0.12 ± 0.02	0.07 ± 0.01	0.09 ± 0.01	0.07 ± 0.01	0.06 ± 0.01	0.08 ± 0.02	0.07 ± 0.01
D5	0.04 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	0.06 ± 0.03	0.07 ± 0.01	0.06 ± 0.02	0.11 ± 0.06	0.14 ± 0.08
D6	0.78 ± 0.02	0.80 ± 0.03	0.63 ± 0.02	0.66 ± 0.02	0.58 ± 0.02	0.47 ± 0.01	0.54 ± 0.02	0.55 ± 0.01
D7	0.24 ± 0.08	0.24 ± 0.08	0.18 ± 0.06	0.19 ± 0.07	0.19 ± 0.07	0.19 ± 0.07	0.24 ± 0.09	0.23 ± 0.07
D8	0.57 ± 0.17	0.56 ± 0.15	0.54 ± 0.14	0.54 ± 0.13	0.53 ± 0.12	0.72 ± 0.17	0.60 ± 0.12	0.59 ± 0.11
D9	0.40 ± 0.05	0.42 ± 0.08	0.38 ± 0.07	0.37 ± 0.05	0.36 ± 0.07	0.38 ± 0.09	0.51 ± 0.06	0.49 ± 0.07

TABLE 8. MSE of regression task with hashing encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.77 ± 0.14	0.81 ± 0.15	0.74 ± 0.13	0.76 ± 0.14	0.69 ± 0.10	0.62 ± 0.09	0.84 ± 0.23	0.79 ± 0.05
D2_p0	0.19 ± 0.03	0.21 ± 0.04	0.02 ± 0.00	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.14 ± 0.02	0.09 ± 0.02
D2_p1	0.19 ± 0.03	0.21 ± 0.04	0.02 ± 0.00	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.14 ± 0.02	0.09 ± 0.02
D3_p0	0.22 ± 0.01	0.24 ± 0.02	0.04 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.06 ± 0.00	0.07 ± 0.00
D3_p1	0.06 ± 0.01	0.06 ± 0.01	0.03 ± 0.00	0.05 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.05 ± 0.00	0.05 ± 0.00
D3_p2	0.23 ± 0.01	0.24 ± 0.01	0.17 ± 0.01	0.19 ± 0.01	0.18 ± 0.01	0.13 ± 0.01	0.16 ± 0.01	0.15 ± 0.01
D4_p0	0.05 ± 0.02	0.06 ± 0.03	0.03 ± 0.01	0.03 ± 0.02	0.02 ± 0.01	0.02 ± 0.00	0.03 ± 0.01	0.03 ± 0.01
D4_p1	0.06 ± 0.02	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.03 ± 0.01
D4_p2	0.06 ± 0.02	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.02 ± 0.01	0.04 ± 0.01	0.03 ± 0.01
D4_p3	0.04 ± 0.02	0.04 ± 0.03	0.02 ± 0.01	0.02 ± 0.02	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01
D4_p4	0.06 ± 0.03	0.06 ± 0.03	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.03 ± 0.01
D4_p5	0.05 ± 0.02	0.06 ± 0.03	0.03 ± 0.02	0.04 ± 0.02	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
D4_p6	0.12 ± 0.02	0.12 ± 0.02	0.07 ± 0.01	0.09 ± 0.01	0.07 ± 0.01	0.06 ± 0.01	0.08 ± 0.02	0.07 ± 0.01
D5	0.04 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.06 ± 0.03	0.07 ± 0.01	0.06 ± 0.02	0.11 ± 0.05	0.14 ± 0.08
D6	0.78 ± 0.02	0.80 ± 0.03	0.65 ± 0.02	0.66 ± 0.02	0.58 ± 0.02	0.47 ± 0.01	0.54 ± 0.02	0.55 ± 0.01
D7	0.24 ± 0.08	0.24 ± 0.08	0.18 ± 0.06	0.19 ± 0.07	0.19 ± 0.07	0.19 ± 0.07	0.25 ± 0.09	0.23 ± 0.07
D8	0.63 ± 0.21	0.60 ± 0.18	0.55 ± 0.15	0.54 ± 0.13	0.55 ± 0.14	0.70 ± 0.17	0.64 ± 0.14	0.59 ± 0.11
D9	0.40 ± 0.05	0.42 ± 0.08	0.41 ± 0.05	0.37 ± 0.05	0.37 ± 0.07	0.39 ± 0.09	0.51 ± 0.06	0.49 ± 0.07

TABLE 9. MSE of regression task with backward difference encoding.

Dataset	LR	SVR linear	SVR Rbf	SVR clinical	GBoosting	RF regressor	KNN	KNN clinical
D1	0.77 ± 0.14	0.81 ± 0.15	0.73 ± 0.13	0.76 ± 0.14	0.69 ± 0.11	0.62 ± 0.09	0.82 ± 0.11	0.79 ± 0.05
D2_p0	0.15 ± 0.02	0.17 ± 0.04	0.06 ± 0.01	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.14 ± 0.02	0.09 ± 0.02
D2_p1	0.15 ± 0.02	0.17 ± 0.04	0.06 ± 0.01	0.05 ± 0.02	0.08 ± 0.01	0.01 ± 0.01	0.14 ± 0.02	0.09 ± 0.02
D3_p0	0.22 ± 0.01	0.24 ± 0.02	0.04 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.07 ± 0.00	0.07 ± 0.00
D3_p1	0.06 ± 0.01	0.06 ± 0.01	0.03 ± 0.00	0.05 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.05 ± 0.00	0.05 ± 0.00
D3_p2	0.23 ± 0.01	0.23 ± 0.01	0.16 ± 0.01	0.19 ± 0.01	0.18 ± 0.01	0.13 ± 0.01	0.15 ± 0.01	0.15 ± 0.01
D4_p0	0.05 ± 0.01	0.06 ± 0.03	0.05 ± 0.00	0.03 ± 0.02	0.02 ± 0.00	0.02 ± 0.01	0.06 ± 0.01	0.03 ± 0.01
D4_p1	0.06 ± 0.02	0.07 ± 0.03	0.05 ± 0.01	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.01	0.03 ± 0.01
D4_p2	0.05 ± 0.01	0.07 ± 0.03	0.05 ± 0.01	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.01	0.03 ± 0.01
D4_p3	0.04 ± 0.02	0.05 ± 0.03	0.03 ± 0.01	0.02 ± 0.02	0.01 ± 0.01	0.01 ± 0.01	0.04 ± 0.02	0.02 ± 0.01
D4_p4	0.06 ± 0.02	0.07 ± 0.03	0.05 ± 0.01	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.01	0.03 ± 0.01
D4_p5	0.05 ± 0.01	0.07 ± 0.02	0.05 ± 0.01	0.04 ± 0.02	0.03 ± 0.01	0.03 ± 0.02	0.06 ± 0.01	0.03 ± 0.01
D4_p6	0.10 ± 0.01	0.11 ± 0.02	0.08 ± 0.01	0.09 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.10 ± 0.02	0.07 ± 0.01
D5	0.04 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	0.06 ± 0.03	0.07 ± 0.01	0.06 ± 0.02	0.11 ± 0.06	0.14 ± 0.08
D6	0.78 ± 0.02	0.80 ± 0.03	0.63 ± 0.02	0.66 ± 0.02	0.58 ± 0.02	0.47 ± 0.01	0.54 ± 0.01	0.55 ± 0.01
D7	0.24 ± 0.08	0.24 ± 0.08	0.18 ± 0.06	0.19 ± 0.07	0.19 ± 0.07	0.19 ± 0.07	0.25 ± 0.09	0.23 ± 0.07
D8	0.56 ± 0.16	0.55 ± 0.14	0.55 ± 0.14	0.54 ± 0.13	0.53 ± 0.12	0.71 ± 0.16	0.60 ± 0.13	0.59 ± 0.11
D9	0.43 ± 0.07	0.45 ± 0.10	0.42 ± 0.05	0.37 ± 0.05	0.36 ± 0.07	0.38 ± 0.08	0.60 ± 0.11	0.49 ± 0.07

In the above figure can be seen the embedded tire space, the reference tire (in red) and its ten closest tires retrieved from the F&M dataset with the clinical kernel used previously in regression tasks.

VII. CONCLUSION

In this work, we proposed a comparison methodology to select the most appropriate ML regression algorithm when dealing with mixed-type variables across multiple datasets. Our approach was presented with a case study of the tire

industry for tire-performance prediction. We have shown that non-parametric Friedman and Nemenyi statistical tests allow us to decide about the appropriateness of encoding strategies for certain classes of regression algorithms. In particular, we showed that ensemble methods, i.e., Random Forest and Gradient Boosting perform significantly better than linear models and KNN and its performance across categorical encodings are equivalent. In contrast, kernel-methods for regression are sensitive to encoding schemes. The SVR linear performance is significantly the worst of the considered

methods with any encoding and SVR Rbf is comparable marginally with ensemble methods when used with binary encoding. Besides, kernels have the benefit of being used for both regression and similarity tasks for data mining applications.

APPENDIX MEAN SQUARE ERROR TABLES

See Table 6–9.

ACKNOWLEDGMENT

The authors would like to thank E. Simon, R. Nava, and M. Ghoniem from LIST as well as L. V. Miegroet, V. Ciorna, and V. M.-Lang from the Goodyear Innovation Center* Luxembourg, for helpful discussion and suggestions.

REFERENCES

- [1] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 23–45, Jan. 2016, doi: [10.1080/21693277.2016.1192517](https://doi.org/10.1080/21693277.2016.1192517).
- [2] M. Awad and R. Khanna, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 1st ed. New York, NY, USA: Apress, 2015.
- [3] B. L. Whitehall and S. C.-Y. Lu, "Machine learning in engineering automation—The present and the future," *Comput. Ind.*, vol. 17, no. 2, pp. 91–100, 1991.
- [4] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20590–20616, 2017.
- [5] A. L. Ferguson, "Machine learning and data science in soft materials engineering," *J. Phys., Condens. Matter*, vol. 30, no. 4, Dec. 2017, Art. no. 043002.
- [6] K. B. Singh and S. Taheri, "Estimation of tire–road friction coefficient and its application in chassis control systems," *Syst. Sci. Control Eng.*, vol. 3, no. 1, pp. 39–61, 2015.
- [7] P. Zhu, F. Pan, W. Chen, and S. Zhang, "Use of support vector regression in structural optimization: Application to vehicle crashworthiness design," *Math. Comput. Simul.*, vol. 86, pp. 21–31, Dec. 2012.
- [8] Y. Ying, S. Xu, J. Li, and B. Zhang, "Compressor performance modelling method based on support vector machine nonlinear regression algorithm," *Roy. Soc. Open Sci.*, vol. 7, no. 1, Jan. 2020, Art. no. 191596, doi: [10.1098/rsos.191596](https://doi.org/10.1098/rsos.191596).
- [9] B. D. Youn, K. K. Choi, R.-J. Yang, and L. Gu, "Reliability-based design optimization for crashworthiness of vehicle side impact," *Struct. Multidisciplinary Optim.*, vol. 26, nos. 3–4, pp. 272–283, Feb. 2004.
- [10] A. Ahmad and S. S. Khan, "Survey of State-of-the-Art mixed data clustering algorithms," *IEEE Access*, vol. 7, pp. 31883–31902, 2019.
- [11] A. H. Foss, M. Markatou, and B. Ray, "Distance metrics and clustering methods for mixed-type data," *Int. Stat. Rev.*, vol. 87, no. 1, pp. 80–109, 2019.
- [12] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle Syst. Dyn.*, vol. 21, no. 1, pp. 1–18, Jan. 1992, doi: [10.1080/00423119208969994](https://doi.org/10.1080/00423119208969994).
- [13] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
- [14] L. Gutiérrez-Gómez and J.-C. Delvenne, "Multi-hop assortativities for network classification," *J. Complex Netw.*, vol. 7, no. 4, pp. 603–622, Aug. 2019.
- [15] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, Feb. 2012.
- [16] K. Singh, P. Sarvari, F. Petry, and D. Khadraoui, *Application of Machine Learning & Deep Learning Techniques in the Context of Use Cases Relevant for the Tire Industry*. Berlin, Germany: VDI Wissensforum, Oct. 2019.
- [17] Y. Yang, L. Sun, and C. Guo, "Aero-material consumption prediction based on linear regression model," *Procedia Comput. Sci.*, vol. 131, pp. 825–831, 2018.
- [18] S. Pang, J. Luo, and Y. Wu, "Properties prediction and design of tire tread composites using machine learning," *Macromol. Theory Simul.*, vol. 29, no. 3, May 2020, Art. no. 1900063, doi: [10.1002/mats.201900063](https://doi.org/10.1002/mats.201900063).
- [19] K. Potdar, T. S., and C. D., "A comparative study of categorical variable encoding techniques for neural network classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, pp. 7–9, Oct. 2017.
- [20] A. Desai, H. Singh, and V. Pudi, "Disc: Data-intensive similarity measure for categorical data," in *Advances in Knowledge Discovery and Data Mining*, J. Z. Huang, L. Cao, and J. Srivastava, Eds. Berlin, Germany: Springer, 2011, pp. 469–481.
- [21] K. Kim and J.-S. Hong, "A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis," *Pattern Recognit. Lett.*, vol. 98, pp. 39–45, Oct. 2017.
- [22] Z. He and Q. Liu, "Deep regression neural network for industrial surface defect detection," *IEEE Access*, vol. 8, pp. 35583–35591, 2020.
- [23] Z. M. Çinar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, p. 8211, Oct. 2020.
- [24] D. Solyali, "A comparative analysis of machine learning approaches for Short-/Long-Term electricity load forecasting in cyprus," *Sustainability*, vol. 12, no. 9, p. 3612, Apr. 2020.
- [25] A. N. Gent and J. D. Walter, *Pneumatic Tire* (Mechanical Engineering Faculty Research), 854 ed. Akron, OH, USA: University of Akron, 2006.
- [26] S. Xu, B. Lu, M. Baldea, T. F. Edgar, W. Wojsznis, T. Blevins, and M. Nixon, "Data cleaning in the process industries," *Rev. Chem. Eng.*, vol. 31, no. 5, pp. 453–490, Jan. 2015.
- [27] H. Liu, S. Shah, and W. Jiang, "On-line outlier detection and data cleaning," *Comput. Chem. Eng.*, vol. 28, no. 9, pp. 1635–1647, 2004.
- [28] D. A. Freedman, *Statistical Models: Theory and Practice*. 2 ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [29] N. S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992, doi: [10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879).
- [30] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.
- [31] A. Daemen and B. De Moor, "Development of a kernel function for clinical data," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 5913–5917.
- [32] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [33] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [34] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
- [35] N. R. Draper and H. Smith, *Applied Regression Analysis*. Hoboken, NJ, USA: Wiley, 2014, ch. 14, pp. 299–325.
- [36] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2009, pp. 1113–1120.
- [37] D. Kaufman and R. Sweet, "Contrast coding in least squares regression analysis," *Amer. Educ. Res. J.*, vol. 11, no. 4, pp. 359–377, Nov. 1974, doi: [10.3102/00028312011004359](https://doi.org/10.3102/00028312011004359).
- [38] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2007, p. 473.
- [39] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.
- [40] H. H. Zhang, *Support Vector Machine Classification for High Dimensional Microarray Data Analysis, With Applications in Cancer Research*. New York, NY, USA: Springer, 2009, pp. 1–24.
- [41] J. Vert, K. Tsuda, and B. Schölkopf, *A Primer on Kernel Methods*, pp. 35–70. Cambridge, MA, USA: MIT Press, 2004.
- [42] T. F. Cox and M. Cox, *Multidimensional Scaling*. 2 ed. Boca Raton, FL, USA: CRC Press, 2000.



LEONARDO GUTIÉRREZ-GÓMEZ received the B.Eng. and B.Math. degrees from the Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, Colombia, the M.Sc. degree in industrial and applied mathematics from the Université de Grenoble 1, France, and the Ph.D. degree from the Université Catholique de Louvain, Belgium, with a dissertation in machine learning on complex networks: dynamical fingerprints, embeddings, and feature engineering. He worked with ST Microelectronics and the Grenoble Computer Science Laboratory (LIG) doing applied research in deep learning, information retrieval, and data mining on graphs. He is currently a Postdoctoral Researcher with the Luxembourg Institute of Science and Technology. His research interests include machine learning in industrial applications, network science, and data mining on graphs.



FRANK PETRY received the Ph.D. and diploma degrees in physics from the University of Heidelberg, in 1993 and 1995, respectively. With the grant support of the Max-Planck Institute for nuclear physics, he continued his research in particle physics. After two years as a Lecturer in applied computer science and algebra, he started at the Product Evaluation Department, Goodyear Technical Center Luxembourg, in 1997. He developed new tire test and modeling methods and published several articles on tire contact mechanics and tire traction. He is currently a Senior Research and Development Associate with the Virtual Capability and Tire/Vehicle Mechanics Department, Goodyear Innovation Center* Luxembourg, Luxembourg.



DJAMEL KHADRAOUI received the master's degree in computer and network security from the École de Mines de Saint-Etienne, France, and the Ph.D. degree in computer vision and robotics from the Université Blaise Pascal. He is currently the Head of the Trusted Service Systems Research Unit, Luxembourg Institute of Science and Technology. His research interests include the design, the security, and the optimisation of service systems enabled by data-intensive infrastructures engineering and aligned with the creation of business impact. He has managed multiple research projects related with the enterprise modeling, value creation, and reliable infrastructures; the data intensive systems with a focus on the processes, the science, and the platforms of big data; the security, privacy, and resilient critical infrastructures with a focus on data privacy and security, cybersecurity, and information security management; and the operations and supply chain optimisation with a focus on data analytics and optimisation, as well as the operations optimisation.

...