Big Data
Volume 00, Number 00, 2021

DOI: 10.1089/big.2021.0067

Explaining Predictive Model Performance: An Experimental Study of Data Preparation and Model Choice

Hamidreza Ahady Dolatsara,[1] Ying-Ju Chen,[2] Robert D. Leonard,[3] Fadel M. Megahed,[3] and L. Allison Jones-Farmer[3,*,i]

Abstract

Although confirmatory modeling has dominated much of applied research in medical, business, and behavioral sciences, modeling large data sets with the goal of accurate prediction has become more widely accepted. The current practice for fitting predictive models is guided by heuristic-based modeling frameworks that lead researchers to make a series of often isolated decisions regarding data preparation and cleaning that may result in substandard predictive performance. In this article, we use an experimental design to evaluate the impact of six factors related to data preparation and model selection (techniques for numerical imputation, categorical imputation, encoding, subsampling for unbalanced data, feature selection, and machine learning algorithm) and their interactions on the predictive accuracy of models applied to a large, publicly available heart transplantation database. Our factorial experiment includes 10,800 models evaluated on 5 independent test partitions of the data. Results confirm that some decisions made early in the modeling process interact with later decisions to affect predictive performance; therefore, the current practice of making these decisions independently can negatively affect predictive outcomes. A key result of this case study is to highlight the need for improved rigor in applied predictive research. By using the scientific method to inform predictive modeling, we can work toward a framework for applied predictive modeling and a standard for reproducibility in predictive research.

Keywords: artificial intelligence; data mining; data science; design of experiments; scientific method; United Network for Organ Sharing (UNOS)

Introduction

In a 1976 article titled *Science and Statistics*, George Box stated "one important idea is that science is a means whereby learning is achieved, not by mere theoretical speculation, on the one hand, nor by undirected accumulation of practical facts on the other, but rather by a motivated *iteration* between theory and practice…. Matters of fact can lead to a tentative theory. Deductions from this tentative theory may be found to be discrepant with certain known or specially acquired facts."[1] A fundamental problem in today's digital economy is how to transform raw data into new knowledge in a way that is more than just an undirected accumulation of facts, but iterates properly through theory and practice so that value-adding learning is achieved.

When analyzing large data sets, individuals/businesses often follow a process that describes the steps needed for knowledge discovery.[2] While the term *data mining* is generally used by news media and industry to denote the entire process,[3] the application of data mining algorithms is a singular step in the knowledge discovery process. Thus, the term *knowledge discovery and data mining (KDDM)* was coined to describe the entire process and avoid the confusion that may arise from how the *data mining* term is used.[2,3] Hereafter we use KDDM to denote the entire process, and use *machine learning (ML)* to denote the step where a statistical/ML algorithm is applied for the purposes of predictive modeling.

When performing KDDM, many follow a structured framework such as the knowledge discovery from

[1]*Graduate School of Management, Clark University, Worcester, Massachusetts, USA.*
[2]*Department of Mathematics, University of Dayton, Dayton, Ohio, USA.*
[3]*Farmer School of Business, Miami University, Oxford, Ohio, USA.*
[i]*ORCID ID (https://orcid.org/0000-0002-1529-1133).*

*Address correspondence to: *L. Allison Jones-Farmer, Department of Information Systems and Analytics, Farmer School of Business, Miami University, Oxford, OH 45056, USA,* E-mail: farmerl2@miamioh.edu

database process,[4] SAS's sampling, exploring, modifying, modeling, and assessing process,[5] and the well-known cross-industry standard process for data mining (CRISP-DM).[6] Many of these frameworks are similar and provide a description of the necessary steps to consider when modeling data. In addition, these frameworks all state the iterative nature of KDDM projects in terms of the need to iterate through steps in the model-fitting process. However, they do not explain *how* to iterate or *when* the user should stop iterating.

In our experience, most implementations of these frameworks resort to trial-and-error experimentation at each decision point in the modeling process. This is like experimentation by changing one factor at a time (OFAT) and translates to researchers and/or practitioners making decisions about the model they are fitting based on *preliminary analyses* that involve a limited number of experimental trials.[7,8] In these preliminary investigations, researchers often focus on *main effects* of changing one aspect of the model fitting process (e.g., whether a random forest [RF] is better than logistic regression [LR]). However, in practice, the superiority of the technique can depend on a combination of prior decisions, which in statistical terms can be denoted by an *interaction effect*. The results obtained from a sequence of OFAT decisions used throughout the KDDM process may be a substandard model. This narrow approach to decision-making in the KDDM process is likely due to a lack of understanding on how these sequences of decisions affect the predictive model performance.

The field of design of experiment (DoE) is centered around the principle that trial-and-error and OFAT approaches to experimentation are suboptimal since they do not: (1) provide practitioners guidance on when to stop iterating; (2) capture possible interactions of the different techniques; and (3) explain why results differ based upon the context of changing a combination of decisions.[9] A DoE framework can be used to enlighten the researcher on the interaction effects between different decisions made in the entire KDDM process. For example, the use of standard interaction plots can be informative in explaining the impact of feature selection approaches on the predictive performance of different ML algorithms. In this article, we use DoE methods to illustrate the combined effect of decisions throughout the KDDM process on the performance of predictive models. Our study is intended to provide guidance to the research community regarding the effect of the decisions they make throughout the model fitting process, from data preparation to model selection.

Specifically, we use a six-factor, full-factorial experimental design applied to a large, publicly available data set to illustrate the effects of (A) four methods of categorical variable imputation; (B) two methods of numerical variable imputation; (C) two categorical variable encoding approaches; (D) five subsampling methods; (E) three feature selection methods; and (F) nine ML algorithms. Our results confirm that the choices we make throughout the KDDM process have an interactive effect on the performance of predictive models. In addition, we give advice to predictive researchers and to those who will evaluate predictive research on ways to explain the effect of decisions made throughout the predictive modeling process.

In the A Bibliometric-Driven Taxonomy of Existing DoE Applications in KDDM section, we use state-of-the-art bibliometric analyses to provide a taxonomy and understand the progression of the literature using experimental design in the KDDM process. In the Application of Experimental Design Methods section, we discuss the methods for our experimental design to study the effect of model-fitting decisions on the performance of predictive models. Next, in the Experimental Design Results section, we provide the results of our experimental study. In the Discussion, Contributions, and Extensions section, we discuss the implications of our experimental study on the model-fitting process. Finally, in the Conclusion section, we provide our concluding remarks.

A Bibliometric-Driven Taxonomy of Existing DoE Applications in KDDM

We are not the first to consider the use of DoE methods in the KDDM process. Several authors have used DoE in the KDDM process to study (1) feature selection applications[10–15] and (2) model parameter tuning applications.[16–38] Figure 1 shows the historical direct citation network[39] with three clusters, for which articles citing a prior work are connected and colored using the same color. In Figure 1, the literature on feature selection applications appears in the four articles near the top of the graph, while the parameter tuning domain is depicted as the complex network at the bottom. We describe the articles within each category in the subsequent paragraphs.

In the feature selection application domain, Kwak and Choi[10] proposed a novel feature selection approach that utilizes Taguchi (orthogonal array) experimental

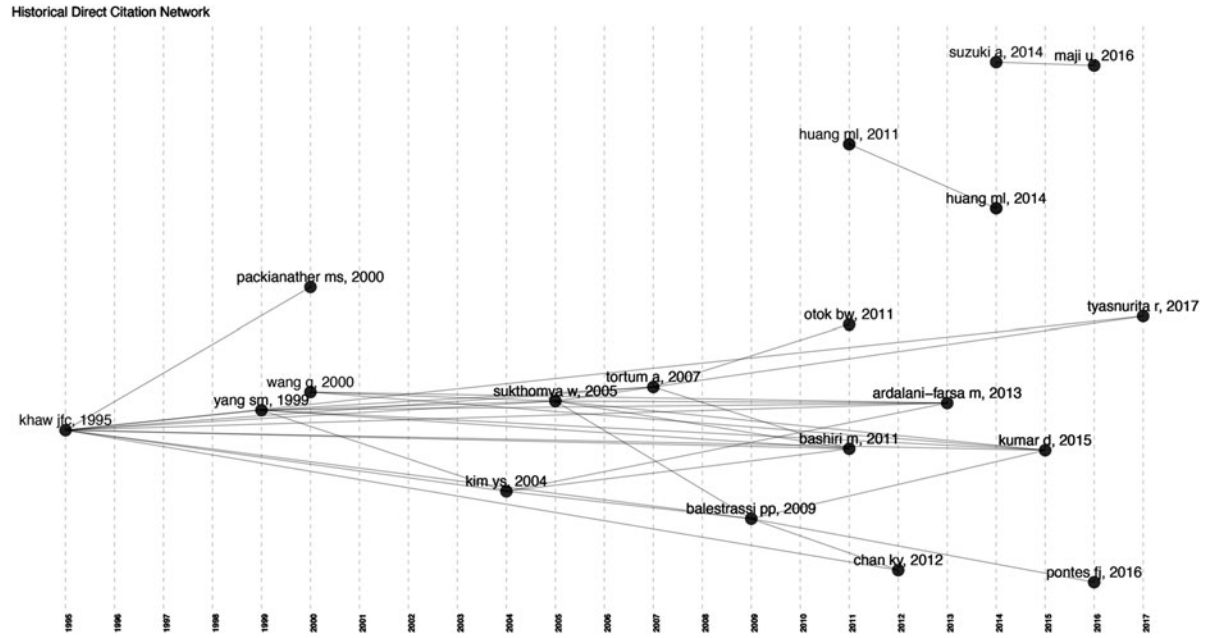

FIG. 1. A bibliometric analysis of the applications of DoE in KDDM projects. DoE, design of experiments; KDDM, knowledge discovery and data mining.

designs to identify "good features with as few experiments as possible." Their approach, denoted as Taguchi method in feature selection (TMFS), was used to select features before the application of a neural network. In their experimental runs, the authors compared the performance of their TMFS approach with another proposed approach based on the *mutual information* criterion as well as combining the feature sets obtained from both proposed methods. Their results showed that the three approaches resulted in good predictive performance when the neural network was applied on their data. The combined approach performed slightly better when compared with the application of a singular feature selection methodology, therefore illustrating an advantage to incorporating the use of experimental design.

Yang et al.[11] proposed a combined feature selection approach that uses genetic algorithms to assist in the *global search* of the feature space and a two-level Taguchi orthogonal array for the *local search*, that is, to determine how to proceed to the next iteration. Their approach, which has a wrapper nature (i.e., is computationally intensive), resulted in an improved classification accuracy in 17 data sets when compared with the baseline condition where the proposed method is not implemented. Other hybrid methods based on Taguchi designs and particle swarm optimization were proposed in Chuang et al.[13] and Allias et al.[14] for selecting the optimal subset of features, which resulted in good performance while reducing the number of features used.

Suzuki and Ryu[12] showed that selecting features based on Taguchi orthogonal arrays, when multivariate regression is used in prediction, results in more accurate predictions when compared with stepwise and no feature selection implementations. Similarly, Maji et al.[15] used Taguchi orthogonal arrays for feature selection when the multiclass naive Bayes is used for classification.

Based on the discussion of feature selection applications, there are three observations to be made. First, four out of the six discussed articles recommended a hybrid approach for feature selection. Second, in each of those articles, only one type of ML algorithm was considered/examined, leaving it unclear whether the authors considered the possibility of their conclusions being different if other algorithms were deployed. Third, the computational time associated with feature selection did not seem to be a constraint for those applications. Fourth, as per the top part of Figure 1, it is interesting to note that the articles discussed in this stream/category have no direct

citations to those of stream 2, which underscores a need for connectedness within the literature. Also note that the two articles of Chuang et al.[13] and Allias et al.[14] are not shown in Figure 1 since they were not available on Web of Science, the resource used to extract the bibliometric information.

In the second application domain, the focus is on how to tune the parameters of either neural networks or support vector machines (SVM) to optimize their prediction performance. Neural network tuning captured the majority of the published articles[16–20,22,23,25–27,30–32,34–38] in this area. In these articles, the Taguchi experimental design methodology was used (with the exception of Bashiri and Geranmayeh[23] who used a central composite design) to optimize neural network designs by concurrently considering the following: (1) microstructure of the network, where decisions are made on the type of transfer function used, learning rules, and/or the representation scheme used for the input layer; and (2) macrostructure of the network, where the number of layers and neurons, and their connection approach are determined. The aforementioned topology is driven by the article of Khaw et al.,[16] which was instrumental to the remainder of the literature as demonstrated in Figure 1. On the contrary, relatively few articles[21,24,28,29,33] considered tuning parameters for SVM by using the experimental design methodology. In these articles, Taguchi designs were used to select the kernel function, its coefficient, and/or the regularization parameter that would result in an "optimal" performance by the SVM.

The research in the two literature streams shown in Figure 1 only considers the potential impact of one step (feature selection or parameter tuning) in the KDDM process on prediction performance. In addition, the analyses were limited to outcomes obtained from a singular ML algorithm. However, researchers typically consider and compare the prediction performance of multiple ML algorithms in their applications. Depending on the problem, researchers also make a sequence of other decisions during the modeling process that include data preparation/cleaning procedures and subsampling strategies to correct for unbalanced cases in the response variable. The effect of these additional decisions, combined with feature selection, model tuning, and ML algorithm, has not been formally considered using DoE methods. Consequently, in this article, we illustrate the effect of these decisions pertaining to the different steps in the KDDM process in *combination* on the effect of the predictive performance of a model.

Application of Experimental Design Methods

In this section, we describe the data and methods used to illustrate the application of DoE methods to quantify the effect of decisions made throughout the KDDM process on the performance of predictive models.

Background and problem description

The United Network for Organ Sharing (UNOS) is a nonprofit association that manages the U.S. organ transplantation system and database.[40] These data were chosen because they are large, publicly available, and have been analyzed by many researchers. To examine the combined effect of decisions made during the modeling process, we consider the 103,570 heart transplantation events recorded in its Organ Procurement and Transplantation Network (OPTN) database between October 1, 1987, and September 30, 2016. This data set presents an excellent illustration for highlighting the interrelations of the different decisions made in the KDDM process since it:

- contains 494 variables, which capture medically important/relevant pre-, intra-, and post-transplant information, therefore requiring a choice of feature selection method;
- has a large number of missing data, which are structurally missing (e.g., UNOS introduced requirements for collecting information on several variables post the starting data of data collection) and/or missing at random. Thus, it is important to evaluate imputation strategies for these missing categorical and numeric variables;
- contains several categorical variables having a large number of possible values (e.g., medical diagnosis codes with 100+ levels), which would require an investigation into how encoding approaches possibly affect prediction performance; and
- is an unbalanced data set as the number of graft survivals for shorter time-periods is much larger than that of graft failures (and *vice versa*), which would require selecting an appropriate resampling strategy.

The Institutional Research Board (IRB) of Miami University has reviewed this study and found that it meets the criteria for exemption according to category 4 of 45 CFR46.

In our analysis, we examine the 1-year graft dichotomous outcome prediction (survival vs. failure) based on using only preoperative variables. With the focus on preoperative variables and the 1-year time frame (as well as removing cases where the response variable is missing or the survival at the 1-year mark is unknown), our UNOS

data set contained 129 predictors (43 numeric and 86 categorical) and 45,005 observations. Furthermore, the imbalance ratio in the response variable is ~ 6.29, which is based on 38,829 observations where the patient has survived at least 1-year post a heart transplantation and 6176 cases where the patient's graft has failed within the 1-year time frame. Note that the 1-year time frame represents one of the most commonly explored time frames involving UNOS data.[41–44]

Experimental design

The overarching question examined in this example is "can we quantify the combined effect of decisions made throughout the KDDM process on the predictive performance of the model?" To address this question, we explore the six factors (and their levels), which we define in Table 1.

In addition to the listed packages, we used the R *caret*[45] package to run all models. Hereafter, we use "model" to denote one combination of factor levels ($A \rightarrow F$) being investigated, that is, a KDDM model and not a specific statistical/ML algorithm. To reduce the computational complexity, we used a random selection of tuning parameter combinations to cover the parameter space for tunable algorithms,[46] setting the number of combinations of algorithm hyperparameter values to 10 for each model (see Table 1 for the hyperparameters needed for each algorithm). Five-fold cross-validation was used for model selection based on the recommendation of James et al.[47] We considered five disjoint test data sets representing five replicates for each unique combination of imputation, encoding, subsampling, feature selection, and algorithm choice/level previously described. We refer the reader to section 3 of the R Markdown document (https://ying-ju.github.io/Explaining-Predictive-Model-Performance.github.io/) for additional details on how the full factorial experiment was conducted (including details on how the UNOS data were preprocessed and how the numeric experiments were conducted using cluster computing).

Measuring model fit

The combination of all factor levels results in 2160 treatment combinations replicated across 5 test samples, resulting in 10,800 experimental runs. One should note that the 2160 ($4 \times 2 \times 2 \times 3 \times 5 \times 9 = 2160$) experiments correspond to the total combinations of all the levels of each factor analyzed. Model fit is measured using the area under the receiver operating characteristic curve (*AUC*). The receiver operating characteristic curve is commonly used in classification analysis and is more informative than accuracy in the case of imbalanced data sets.[48] For a given model, it is a plot of the true-positive rate versus the false-positive rate for many different decision thresholds. The *AUC* measures how well the model distinguishes the positive class from the negative class. An $AUC > 0.5$ indicates the model is better than chance at distinguishing the positive class from the negative class.

Analysis method

The hierarchical regression analysis will be used to analyze the data from this experiment. The method of least squares will be used to estimate all parameters. Here, our response variable is the *AUC*, and we are evaluating the ability of each experimental condition to predict the outcome of 1-year survival in the UNOS data set. To best reflect the sequential nature in which an analyst fits a predictive model, the factors will be entered into the study in four steps as follows so that the additional variability accounted for by the factors entered can be evaluated. Overall, six main effects (factors $A \rightarrow F$ in Table 1) and 15 interaction terms will be evaluated in the following order:

- Analysis 1: *categorical imputation*, *numerical imputation*, *categorical encoding*, and the two-factor (pairwise) interactions among these three factors;
- Analysis 2: *subsampling method* and the two-factor interactions among *subsampling method* and factors in Analysis 1;
- Analysis 3: *feature selection approach*, *algorithm*, and the two-factor interactions among these two factors and the factors in Analyses 1 and 2;
- Analysis 4: *reduced model*—the full model from Analysis 3 will be reduced using the practical significance of the effects.

Analysis 1 represents the decisions that an analyst would make while initially cleaning the data before analysis. After cleaning the data, the analyst would then subsample the data to correct for unbalanced response classes; thus, subsampling and the associated interactions are entered in Analysis 2. Finally, feature and algorithm selection are often completed in tandem. We enter feature selection and algorithm along with their associated two-factor interactions into Analysis 3 simultaneously to evaluate their effect on predictive performance. Once all variables and interactions are considered, the effects are reduced in Analysis 4,

Table 1. The six factors examined in our experimental design

Factor	*Levels*	*Description*	*Reference*
(A) Categorical imputation	No imputation (drop)	Observations containing missing values of categorical variables are dropped	—
	Mode	Missing values of categorical variables are replaced with mode category	—
	Missing	Missing values of categorical variables are labeled as *missing, a new category*	—
	Unknown	Missing values of categorical variables are labeled with the existing *unknown* category	—
(B) Numerical imputation	No imputation (drop)	Observations containing missing values of numerical variables are dropped	—
	Median	Missing values of numerical variables are replaced with the median	—
(C) Encoding	Label	Categorical variables are encoded with numerical values, e.g., 1, 2, 3.	—
	One-hot	Categorical variables are dummy coded	—
(D) Subsampling	None	Leave training data imbalanced	—
	Down	Randomly subset all the majority class in the training set so that its class frequency matches the least prevalent class	—
	Up	Randomly over sample the minority class to match the majority class in the training set	—
	SMOTE	Application of the Synthetic Minority Oversampling TEchnique by utilizing a k-nearest neighbor algorithm to create synthetic data to handle the class imbalance problem	Chawla et al.[64]
	ROSE	Application of the Random Oversampling Examples methodology by developing the feature space of minority and majority class examples that are drawn from a conditional kernel density estimate of the two classes to handle the class imbalance problem	Menardi and Torelli[65]
(E) Feature selection	FFS	Fast Correlation-Based Feature Selection, which selects the subset of informative features according to the correlation between features and class, and between features themselves, implemented with the "Biocomb" R package with the default discretization method MDL and the threshold 0.01	Yu and Liu[66]
	LASSO	Using LASSO for feature selection, which selects the features that are strongly associated with the response variable by shrinking the coefficients of relative insignificant features to 0 through fitting a generalized linear model using penalized maximum likelihood, implemented with "glmnet" R package with loss, AUC, to use for 10-fold cross-validation	Tibshirani[67]
	RF	Utilizing random forest as an embedded feature selection method that evaluates the importance of each variable by ranking how well the tree-based decisions used by random forests improve the purity of the node, implemented with "ranger" R package with the feature importance option: permutation, the number of variables to possibly split at in each node: 20% of the number of the predictors, the minimal node size: 10% of the data size, the number of threads: 1	Chandrashekar and Sahin[68]
(F) Algorithm	ANN	A neural network implemented with the "nnet" R package with two tuning parameters: # hidden layers and weight decay	—
	DT	A CART decision tree, implemented based on the "rpart" R package, with one tuning parameter: complexity parameter	—
	ElasticNet	An elastic net from the "glmnet" R package with two tuning parameters: mixing % and regularization parameter	—
	KPLSR	Kernel partial least square regression, implemented based on the "pls" library, with the # of components as its only tuning parameter	Rosipal and Trejo[69]
	LDA	Linear discriminant analysis, using the "lda" method from the "MASS" package. No tune parameters are available for this method	—
	LR	Logistic regression implementation using "glm" from base R. No tune parameters are available for this method	—
	NB	Naive Bayes from the "naivebayes" package with Laplace correction, distribution type, and bandwidth adjustment parameters	—
	RF	Random forest implemented using the "ranger" package with # of predictors, splitting rule, and minimal node size parameters	—
	XGB	eXtreme Gradient Boosting, which is a scalable and efficient implementation of Friedman et al.[70] using the "xgbDART" method from the "xgboost" package. Method has nine tuning parameters	Chen and He[71]

retaining only the factors and interactions that explain a significant proportion of the variability in model fit as measured by *AUC*. Details regarding the factors and levels included in each analysis are provided in Table 1.

Experimental Design Results

The experimental design resulted in 2160 experimental conditions in the form of ML models to predict the outcome variable of 1-year survival for the UNOS data set. Using the 5 test data sets, this resulted in 10,800 experimental runs. The 10,800 ML models were fitted to the UNOS data set using an Intel® Xeon® processor-based supercomputer containing 23,392 CPU cores, with a typical 28 cores/node setup and memory per node of 128 GB. Note that the access to the supercomputer center made our experimental design feasible (which as mentioned earlier is only intended for quantifying the impact of the different decisions on prediction

accuracy and should not be expected in a typical ML study). Seventeen cases in the experiment did not converge, resulting in 10,783 cases available for analysis. All 17 cases of nonconvergence occurred with the Kernel Partial Least Square Regression algorithm when no subsampling strategy was used. A complete detail of the cases that did not converge is provided in section 3 of the R Markdown document (https://ying-ju.github.io/Explaining-Predictive-Model-Performance.github.io/).

Table 2 summarizes the hierarchical regression results for each analysis. Six statistics are reported: (1) R^2, the coefficient of determination, is the proportion of the variation in the values of the response variable (AUC) that is explained by the linear regression model of the responsible variable (AUC) on the predictors (factors). This value is always between 0 and 1. A value of 1 means the relationship is perfectly linear and the regression line perfectly predicts the observation, while a value of 0 means there is no linear relationship, and the regression model does a very poor job. (2) ΔR^2 is the difference of R^2 values between two analyses. (3) ΔF is the test statistics based on a Wald test for each two consecutive analyses. (4) df_1 is the difference in degrees of freedom between two analyses. (5) df_2 is the residual degrees of freedom. (6) SigΔF is the p-value for a Wald test for each two consecutive analyses. Note that Analyses 1–3 are defined in the Analysis Method section. Furthermore, Analysis 4 was reduced from Analysis 3, and contains categorical imputation, numerical imputation, subsampling, feature selection, and algorithm main effects, and the following two-way interaction terms: numerical imputation × categorical imputation; numerical imputation × feature selection; subsampling × features selection; subsampling × algorithm; and feature selection × algorithm. In Analysis 1, we see that numerical and categorical imputation methods as well as the encoding method for categorical variables account for only 1% of the observed variability in the measure of fit, *AUC*, in this experimental study ($R^2 = 0.01$). More details about Analysis 1 can be found in Table 3. In Table 3, we see that numerical and categorical imputation methods interact to significantly estimate the *AUC* ($F_{3,10770} = 14.07$, $p = 0.000$) for the models used to predict 1-year survival in the UNOS data set.

Table 2. Summary of the four analyses

Analysis	R^2	ΔR^2	ΔF	df_1	df_2	*Sig*ΔF
1	0.010					
2	0.254	0.244	146.25	24	10,770	0.000
3	0.894	0.640	554.32	116	10,746	0.000
4	0.890	−0.004	5.33	73	10,703	0.000

Table 3. Type II sum of squares and analysis of variance *F*-tests for Analysis 1

Analysis 1	*Type II sum of squares*	df	F *value*	Pr(>F)
Numerical imputation	0.07	1	17.868	0.000
Categorical imputation	0.17	3	15.858	0.000
Encoding	0.00	1	0.025	0.874
Numerical imputation × categorical imputation	0.15	3	14.067	0.000
Numerical imputation × encoding	0.00	1	0.187	0.666
Categorical imputation × encoding	0.00	3	0.330	0.804
Residuals	39.36	10,770		

In Analysis 2, the main effect for subsampling is added along with all two-factor interactions that involve subsampling. The overall R^2 for Analysis 2 increases to 0.254. There is a significant difference in the variation in *AUC* accounted for by the inclusion of subsampling and the two-factor interactions in Analysis 2 compared with Analysis 1 as indicated by $\Delta R^2 = 0.244$ ($\Delta F_{24,10770} = 146.25$, $p = 0.000$). Investigation of Table 4 shows that the interaction between numerical and categorical imputation methods is significant ($F_{3,10746} = 18.49$, $p = 0.000$), and the main effect for subsampling method is also statistically significant ($F_{4,10746} = 875.01$, $p = 0.000$) in estimating the *AUC*. Recall that subsampling is used to achieve balance when there are unbalanced cases in a categorical response variable as is the case with the survival response in the UNOS data.

In Analysis 3, the main effects for feature selection and algorithm along with all two-way interactions that involve these factors are added to the models. The overall R^2 increases to 0.894. There is a significant difference in the variation in *AUC* accounted for by the inclusion of

Table 4. Type II sum of squares and analysis of variance *F*-tests for Analysis 2

Analysis 2	*Type II sum of squares*	df	F *value*	Pr(>F)
Numerical imputation	0.066	1	23.768	0.000
Categorical imputation	0.173	3	20.889	0.000
Encoding	0.000	1	0.029	0.864
Subsampling	9.664	4	875.010	0.000
Numerical imputation × categorical imputation	0.153	3	18.485	0.000
Numerical imputation × encoding	0.001	1	0.250	0.617
Numerical imputation × subsampling	0.007	4	0.665	0.616
Categorical imputation × encoding	0.004	3	0.437	0.726
Categorical imputation × subsampling	0.017	12	0.507	0.912
Encoding × subsampling	0.003	4	0.291	0.884
Residuals	29.672	10,746		

Table 5. Type II sum of squares and analysis of variance *F*-tests for Analysis 3

Analysis 3	*Type II sum of squares*	df	F *value*	Pr(>F)
Numerical imputation	0.065	1	163.584	0.000
Categorical imputation	0.172	3	144.492	0.000
Encoding	0.000	1	0.211	0.646
Subsampling	9.666	4	6102.520	0.000
Feature selection	0.713	2	900.368	0.000
Algorithm	5.657	8	1785.642	0.000
Numerical imputation × categorical imputation	0.153	3	128.397	0.000
Numerical imputation × encoding	0.001	1	1.658	0.198
Numerical imputation × subsampling	0.008	4	4.792	0.001
Numerical imputation × feature selection	0.097	2	122.576	0.000
Numerical imputation × algorithm	0.050	8	15.628	0.000
Categorical imputation × encoding	0.004	3	3.059	0.027
Categorical imputation × subsampling	0.017	12	3.513	0.000
Categorical imputation × feature selection	0.045	6	19.149	0.000
Categorical imputation × algorithm	0.012	24	1.271	0.169
Encoding × subsampling	0.003	4	2.024	0.088
Encoding × feature selection	0.000	2	0.065	0.937
Encoding × algorithm	0.015	8	4.762	0.000
Subsampling × feature selection	0.103	8	32.445	0.000
Subsampling × algorithm	17.736	32	1399.689	0.000
Feature selection × algorithm	1.031	16	162.745	0.000
Residuals	4.209	10,630		

feature selection method and algorithm along with the two-factor interactions in Analysis 3 compared with Analysis 2, as indicated by $\Delta R^2 = 0.640$ ($\Delta F_{116,10746} = 554.32$, $p = 0.000$). From Table 5, it is clear that the inclusion of feature selection method and algorithm resulted in a number of significant interaction terms, many of which are associated with small Type II Sum of Squares values. The large number of error degrees of freedom (10,630) in this analysis renders even small effect sizes statistically significant, and thus, we evaluate the size of the effects in Analysis 3 using partial η^2, denoted η_p^2.

When analysis of variance is used for explanatory analysis, η_p^2 is a common measure of effect size. Popularized by Cohen,[49] the effect size as measured by η_p^2 for a given factor, A, is given by the following:

$$\eta_p^2 = \frac{SS(A)}{SS(A) + SS(error)},$$

where $SS(\cdot)$ is the Sum of Squares. The use of η_p^2 has been controversial due to inaccurate reporting and improper explanation of its meaning in some areas of the behavioral sciences.[50] Although Cohen[49] suggested heuristic thresholds for small, medium, and large effect sizes based on the size of η_p^2, these thresholds are largely based on intuition, rather than empirical or statistical results.

For this analysis, we report η_p^2 as a measure of the variance due to a given factor (or interaction) effect relative to the sum of the variance due to the factor (or interaction) and the unexplained error. The η_p^2 values for all factors and interaction terms in Analysis 3 are given in Table 6. We use these values to reduce Analysis 3, retaining only effects associated with $\eta_p^2 > 0.02$. In addition, we retained any main effect that was involved in an interaction effect. This is an admittedly arbitrary threshold that retains the 10 effects that are practically meaningful in this study (see gray-shaded rows in Table 6). We provide the data from our experiment and the code for the full analysis in section 4.3 of the R Markdown document (https://ying-ju.github.io/

Table 6. Partial η^2 from the effects of Analysis 3

	Partial η^2
Numerical imputation	0.015
Categorical imputation	0.039
Encoding	0.000
Subsampling	0.697
Feature selection	0.145
Algorithm	0.573
Numerical imputation × categorical imputation	0.035
Numerical imputation × encoding	0.000
Numerical imputation × subsampling	0.002
Numerical imputation × feature selection	0.023
Numerical imputation × algorithm	0.012
Categorical imputation × encoding	0.001
Categorical imputation × subsampling	0.004
Categorical imputation × feature selection	0.011
Categorical imputation × algorithm	0.003
Encoding × subsampling	0.001
Encoding × feature selection	0.000
Encoding × algorithm	0.004
Subsampling × feature selection	0.024
Subsampling × algorithm	0.808
Feature selection × algorithm	0.197

The gray shaded rows indicate effects retained for Analysis 4.

Table 7. Type II sum of squares and analysis of variance *F*-tests for Analysis 4

Analysis 2	*Type II sum of squares*	df	F *value*	Pr(>F)
Numerical imputation	0.065	1	158.661	0.000
Categorical imputation	0.172	3	140.381	0.000
Subsampling	9.667	4	5927.912	0.000
Feature selection	0.713	2	874.381	0.000
Algorithm	5.658	8	1734.822	0.000
Numerical imputation × categorical imputation	0.152	3	124.332	0.000
Numerical imputation × feature selection	0.097	2	118.763	0.000
Subsampling × feature selection	0.103	8	31.493	0.000
Subsampling × algorithm	17.736	32	1359.500	0.000
Feature selection × algorithm	1.031	16	158.080	0.000
Residuals	4.363	10,703		

Explaining-Predictive-Model-Performance.github.io/) so that the interested reader can explore the effects and interaction terms that would be retained (or eliminated) with other cutoff values for η_p^2.

From Table 2, the Analysis 4 $R^2 = 0.89$ has decreased by only $\Delta R^2 < 1\%$, although the difference between Analysis 3 and 4 is statistically significant ($\Delta F_{79,10703} = 5.33$, $p = 0.000$). Table 7 shows that all main effects and two-factor interaction terms included in Analysis 4 are statistically significant. Figure 2 provides the two-way interaction plots from Analysis 4 with error bars indicating 95% prediction intervals on each marginal mean.

Discussion, Contributions, and Extensions

Lessons learned from the heart transplantation case study

The purpose of case study analysis is to illustrate how decisions made throughout the model-fitting process can affect the predictive performance of a model. Using a carefully designed experiment, we have illustrated that the decisions made in data preparation and model fitting interact to affect the predictive performance of models fit to a large publicly available, and widely modeled database. Although the conclusions from this study are specific to the data analyzed here, the results illustrate how early decisions in the modeling process can interact with later decisions in ways that may be unexpected by researchers and peer reviewers.

Our study design and results differ from hyperparameter optimization using a grid search approach (see, e.g., GridSearchCV in Scikit-Learn[51]) or Bayesian optimizers.[52] Our study considers the effect of many decisions that occur both before and during the KDDM process that would not be considered in these grid search approaches. In this study, we attempted to include many of the decisions from the beginning (data preparation) to the end (feature and algorithm selection) of the model building process within the KDDM framework.

When fitting a predictive model, data cleaning is often completed in one stage, before model fitting. From Figure 2A, for the UNOS data, dropping missing values in both categorical and numerical variables seems to have a very small effect in the predictive models compared with the results based on the other imputation strategies. In addition, dropping missing values in numerical variables, but imputing missing values in categorical variables, results in predictive models with the slightly higher AUC values compared with the AUC values based on the approaches of replacing missing values with the median in numerical variables without dropping missing data in categorical variables. Interestingly, our results show that for the UNOS data, there is a small, but significant, interaction between the method of imputation used for missing data in numerical variables and the feature selection method, which often occurs much later in the modeling process. From Figure 2B, we see that when there are missing data on the numerical variables, the RF method of feature selection resulted in predictive models with slightly lower AUC values. In general, replacing the numerical missing values with the median and using either Fast Feature Selection or LASSO for variable selection resulted in the highest AUC in our study. Without knowledge of the interactive relationship between imputation methods and feature selection, a researcher is likely to make data cleaning decisions independent of model fitting decisions.

Another important lesson learned from this analysis is the importance of the subsampling strategy to manage the unbalanced cases. One strategy, random oversampling examples (ROSE), worked particularly poorly in many cases, interacting significantly with the feature selection method (Fig. 2C) and algorithm (Fig. 2D). The ROSE subsampling strategy worked poorly with all methods of feature selection, resulting in AUC values below 0.5, on average in all cases. From Figure 2D we see that the predictive performance of most algorithms is relatively robust to the choice of the subsampling strategy with the exception of a few combinations, including the use of the ROSE method with artificial neural network (ANN), ElasticNet, and LR.

Although feature and algorithm selection is often completed in tandem, better performance can often be obtained by matching the feature selection method

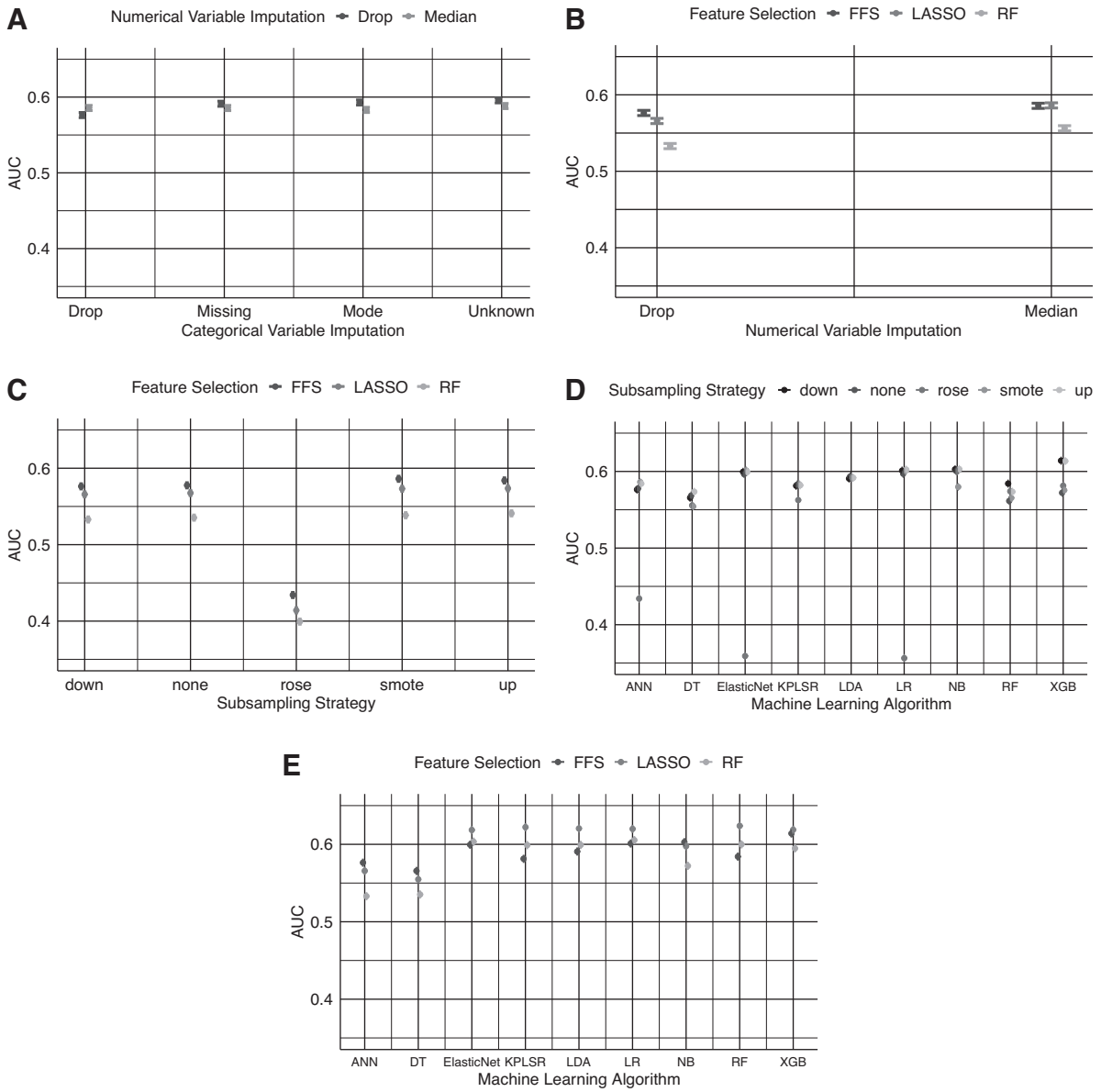

FIG. 2. Plots of the two-factor interactions included in Analysis 4. **(A)** Numerical × categorical imputation interaction; **(B)** feature selection × numerical imputation; **(C)** feature selection × subsampling interaction; **(D)** subsampling × algorithm interaction; and **(E)** feature selection × algorithm interactions. The interested reader is referred to section 4.3.7 of our Markdown document (https://ying-ju.github.io/Explaining-Predictive-Model-Performance.github.io/) for an enlarged view of each subfigure. ANN, artificial neural network; AUC, area under the receiver operating characteristic curve; DT, Decision Tree; FFS, Fast Correlation-Based Feature Selection; KPLSR, kernel partial least square regression; LASSO, LASSO regression; LDA, linear discriminant analysis; LR, logistic regression; NB, naive Bayes; RF, random forest; XGB, eXtreme Gradient Boosting.

with the algorithm as shown in Figure 2E. Interestingly, for most algorithms, the highest average AUC values were seen with models that used the LASSO method of feature selection. The Fast Correlation-Based Feature Selection method of feature selection is preferred for ANN and Decision Tree algorithms and illustrates that decisions about feature selection should be taken into consideration by the algorithm that is used.

Extending our approach to other problems: a potential framework for statistical experiments within the KDDM process

In early sections, we have focused on examining how statistically designed experiments can inform the decisions made in the KDDM process in the context of a large, complex, and widely cited heart transplantation data set. That being said, one cannot expect the results of our analysis to be generalizable to other data sets (especially since we have chosen to deploy a large-scale computational experiment for a singular data set). Hence, in this subsection, we attempt to provide a framework that can be used by researchers/practitioners in other applications.

The process of planning, conducting, and analyzing a designed experiment can be divided into seven phases.[9] Phase 1 pertains to formulating the problem, identifying relevant data, graphing/performing exploratory analysis of the data, and identifying the goals of the experiment. In Phase 2, one would determine the experimental factors and their levels. From a nonstatistical perspective, these constitute the decisions that should be evaluated/optimized by the experimental protocol. Phase 3 involves the selection of an appropriate performance metric (e.g., the choice of accuracy vs. the AUC). Thus, the first 3 phases refer to pre-experimentation planning. Phase 4 involves choosing an appropriate statistical design for the experiment, which includes accounting for the availability of computational resources. In Phase 5, the experiment is conducted. That is, the KDDM is executed for the experimental conditions defined in the fourth phase. The analysis of the experimental data is performed in Phase 6, which is followed by conclusions and recommendations for deploying a specific KDDM model and/or running follow-up experiments in Phase 7. A visual summary of how the experimental design procedure can be adopted for augmenting the decision-making in the KDDM process is shown in Figure 3.

While it is obvious that any KDDM project would involve some level of planning, Phase 1 ensures that a project charter is developed and agreed upon by different team members (often representing different business units). In our opinion, there are two perspectives that should be incorporated in defining the problem. First, one should incorporate the existing business/engineering perspective in: (1) scraping/querying all relevant data, (2) performing exploratory data analysis on the scraped data, (3) determining the computational resources that can be allocated to the project, and (4) identifying how the project should be executed to account for the existent project infrastructure within the firm (e.g., should the project follow an agile methodology) and the immediacy level for completion of the project. The reader is referred to the KDDM and/or CRISP-DM processes for additional information on what can be incorporated as a part of this analysis. The second, and often ignored perspective, is the statistical perspective, which allows for describing the purpose of using experimentation to augment the decision-making process within the KDDM. Specifically, it is often helpful to describe whether the purpose of the experiment is: (1) screening, that is, understanding how the different factors contribute to the variation in the response of interest, or (2) optimization, where the team has a good understanding of the system being studied, and thus, the goal is to maximize the performance metric of choice (e.g., AUC).

The choice of the experimental factors and their number of levels depends on both the objectives specified in Phase 1 and the nature of the data. In Phase 2, we suggest seven possible factors and some of their potential levels. If the data set contains missing data, one has to consider how to handle such data, which can include excluding observations with missing data, categorical and/or numerical imputation techniques, and stochastic/multiple imputation techniques. Since categorical and numerical imputations are often considered separately,[41,44,53] we have presented them as the first and second factors in Figure 3. The third factor, categorical variable encoding, is required for many statistical models (e.g., LR) and in specific implementations of ML models (e.g., the classification and regression tree model by Breiman et al.[54] that limits the number of levels of a factor to ≤32). Possible levels of encoding include the following: (1) label encoding, where the variable is treated as a numeric variable to facilitate the analysis, and (2) one-hot encoding, where $k - 1$ binary variables/features are generated to correspond to the k categories of the variable. We encourage researchers and practitioners to graph their target variable against the factor levels for both nominal and

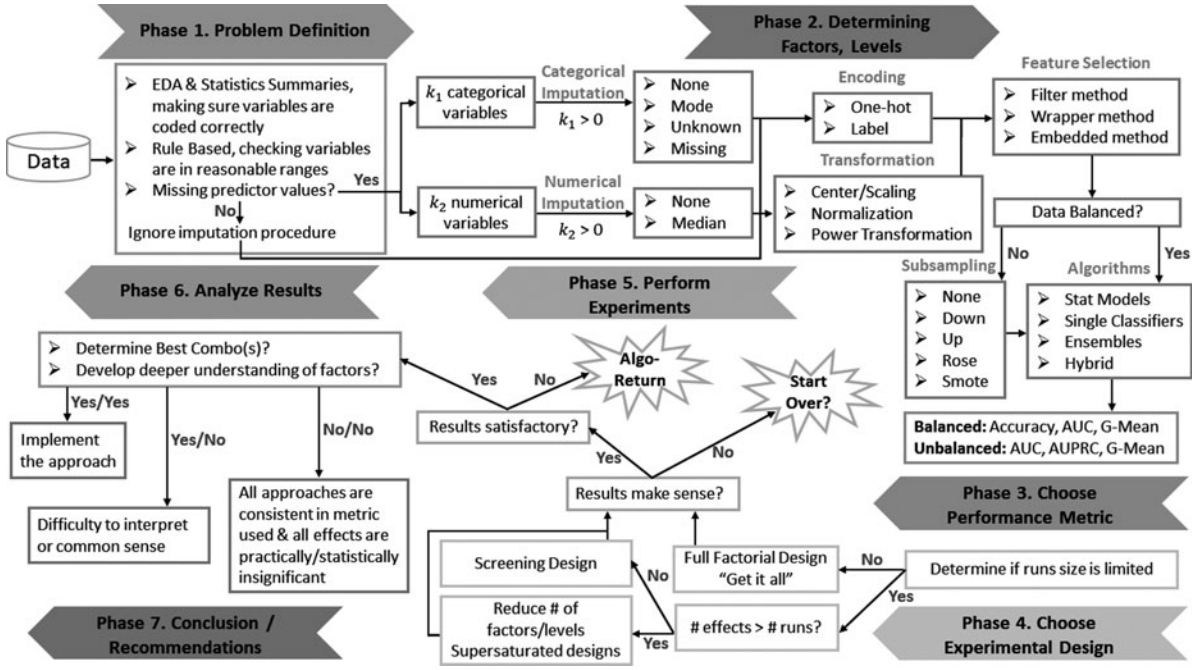

FIG. 3. An overview of how experimental design can be used to inform the decisions made within the KDDM process. AUPRC, area under the precision recall curve.

ordinal variables before modeling if possible. We believe this will help to determine the best encoding scheme for a given data set. A fourth factor can be the application of transformations to one or more of the predictor variables. These transformations can include centering, scaling, normalization, logarithmic, and/or Box-Cox transformations. While these are known to improve the performance of linear models, we have not considered them in our experiment to reduce the number of experimental runs. That being said, it is recommended to normalize predictors before the application of the classification models.[55] The fifth factor is the impact of feature selection, which is typically considered in most ML applications. Possible levels here include different feature selection methods within filter, wrapper, and embedded techniques. A sixth possible factor is the utilization of subsampling techniques if the categorical response variable is imbalanced. Possible factor levels include the following: upsampling the minority class, downsampling the majority class, and the application of the synthetic majority oversampling technique. The seventh factor involves decisions to be made about the ML algorithms and their tuning. Since some approaches, such as LR, do not require tuning, we recommend that different implementations of a given algorithm be considered as different factor levels here. For example, in one application, we can consider the levels of this factor to include the following: (1) LR, (2) ANN with a sigmoid activation function, and (3) ANN with the hyperbolic tangent function as an alternate activation function.

In Phase 3, the focus is on selecting an appropriate performance metric to evaluate the KDDM process. The choice of performance metric should be accounted for by the following: (1) the type of response variable being evaluated by the competing ML algorithms since the prediction performance of models for dichotomous, categorical, and numeric response variables is evaluated differently, and (2) in the case of dichotomous responses that are imbalanced, metrics such as the AUC and the area under the precision recall curve are more appropriate than *accuracy*.

The choice of an experimental design in Phase 4 depends on the outcomes from Phases 1 to 3. There are two main scenarios that will dictate the choice of the experimental design. Scenario 1 corresponds to the case when there are no restrictions pertaining to the number of experimental runs/trials that can be evaluated. This may be an unreasonable assumption as most practitioners will not have access to a supercomputer to evaluate

the joint effect of multiple factors on a particular data set. Scenario 2 corresponds to the more common scenario where running all possible combinations of factor levels is not feasible. All of the articles cited in the A Bibliometric-Driven Taxonomy of Existing DoE Applications in KDDM section use orthogonal arrays, which are a subset of full factorial designs in which all possible experimental conditions are evaluated. An alternative to full factorial designs is optimal designs. Optimal designs are created by using an interchange algorithm to update the factor-level combinations that define each row of the experimental design for some optimality criteria. For KDDM experiments, we recommend the commonly used *D*-optimality criterion[56] since it focuses on an ability to estimate a specified subgroup of effects (e.g., easy to interpret effects such as main effects and two-factor interactions).

Once designed, the experiment is carried out in Phase 5 according to the specified plan formed while progressing through Phases 1–4. The goals of this phase are threefold: (1) executing the designed experiment as planned, (2) providing implementable code/procedures such that experimental results can be obtained, and (3) storing the intermediate and final results from the experiment such that they can be made available for further analyses.

The analysis in Phase 6 presents a different paradigm than what is typically done in KDDM projects. Specifically, as illustrated in Figure 3, Phase 6 has two main goals: (1) determining the best combination of factor levels that maximize the performance metric of choice (traditional goal of any KDDM project), and (2) modeling how the factors of interest contribute to the obtained values of the performance metric (traditional goal of experimental design applications). Note that the latter goal allows for understanding the possible influences of interaction effects, which is one of the main advantages of incorporating a designed experiment into the overall methodology of investigating ML algorithms.

As illustrated in Figure 3, Phase 7 is where conclusions are drawn based on an analysis of the data and recommendations made for what steps to take next. The recommendations can be categorized according to the following possible outcomes from the analysis: (1) implement the best approach if results show a specific combination of factor levels leading to a superior performance; (2) implement the simplest approach if there are no significant differences in the values of the chosen performance metric; and (3) perform a follow-up experiment if neither of the previous two conditions is met or it is believed that additional useful information can be gathered with another experiment. The reader should refer to Montgomery[9] for more details on how a follow-up experiment should be designed.

Conclusion

The overarching motivation of this work is to posit the need for improved rigor to predictive research. Our approach to analyzing the provided case study attempts to capitalize on our training/indoctrination in the *scientific method* and examine how it can be utilized to inform the *art* of predictive analytics. An immediate outcome from the case study is showing that it is possible to quantify the impact of the decisions made throughout the entire KDDM process (as opposed to just hyperparameter optimization) on the final model's predictive accuracy (which in our case study was measured using AUC). We highlight the following important building blocks of using experimental design as a basis for a more scientific approach to existing KDDM frameworks:

Explanatory predictive modeling

Recently, ML researchers have started to realize that a significant obstacle toward successful implementations in many applications is the lack of practitioner trust and/or regulations that may limit the adoption of *black box* predictive models. Thus, an emerging and growing theme in the ML research community is developing approaches for *interpretable machine learning*.[57–60] These approaches focus on attempting to understand the impact of selected features on the model's prediction, identifying when an algorithm should not be trusted, and providing guidance on computing feature importance for different algorithms. Therefore, our approach can be considered complementary to those approaches, where we focus instead on how the decisions made throughout the KDDM process impact the predictive accuracy of the obtained models. As researchers, it is important to consider all of the decision points throughout the KDDM process and how they affect the predictive performance of a model.

Establishing a reproducibility standard

The inability to replicate research results has received significant attention in social and medical sciences.[61] In the context of KDDM, we need to distinguish between *reproducibility* and *replicability*. These two terms are succinctly defined by Peng[62(p.31)] as follows:

> *Replication* is the cornerstone of scientific research…. The replicability of a study is related to the chance that an independent experiment targeting the same scientific question will produce a result consistent with the original study…. Recently, a variation of this concept, referred to as reproducibility, has emerged as a key minimum acceptable standard, especially for heavily computational research.
> *Reproducibility* is defined as the ability to recompute data analytic results, given an observed data set and knowledge of the data analysis pipeline.

Unfortunately, even the minimum standard of *reproducibility* in computational research is frequently not met.[63] In our estimation, this *reproducibility crisis* in computational research is, in most cases, not due to *foul play*. Instead, it arises as a result of poor documentation of the many intricacies involved in computational data-driven research [e.g., refer to how our R Markdown document (https://ying-ju.github.io/Explaining-Predictive-Model-Performance.github.io/) has captured information that was hard to explain/show in section 3]. Fortunately, the solution is simple, authors should provide their commented code, data, and ideally their analysis using Markdown/Jupyter notebooks. Nowadays, platforms such as Zenodo (https://zenodo.org/) allow one to obtain unique document identifiers for one's code and data.

In conclusion, there are several opportunities for improving the rigor and reporting of predictive research using KDDM processes. This article highlights how the decisions made throughout the data preparation and model-fitting process can interact to affect the predictive performance of a model.

Supplementary Information

Data availability

The data used in this study can be requested from UNOS at: https://optn.transplant.hrsa.gov/data/request-data/. The data are freely available for members of the *OPTN* and can be purchased by members of academic institutions and/or the public. In the data request, the reader should select the "STAR" file. In this study, we used all data up to September 30, 2016.

Code and analysis

The results of data analysis and code are available online at https://ying-ju.github.io/Explaining-Predictive-Model-Performance.github.io/.

Authors' Contributions

Conceptualization: Y.-J.C and F.M.M. Computational work: Y.-J.C. and H.A.D. Statistical analysis: L.A.J.-F. and R.D.L. Writing—original: Y.-J.C., H.A.D., L.A.J.-F., R.D.L., and F.M.M. Reviewing and editing: Y.-J.C., L.A.J.-F., and F.M.M.

Acknowledgment

The UNOS registry (our data provider) was supported, in part, by the Health Resources and Services Administration contract 234-2005-37011C.

Disclaimer

The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does the mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

Author Disclosure Statement

The authors declare no conflict of interest.

Funding Information

The modeling approach, analysis, and computational resources were supported, in part, by the Ohio Supercomputer Center (Grant Nos. PMIU0138, PMIU 0162, and PMIU0166) and the National Science Foundation (Grant No. CMMI-1635927). Dr. Jones-Farmer's research was partially supported by the Van Andel Professorship at Miami University.

References

1. Box GEP. Science and Statistics. J Am Stat Assoc. 1976;71:791–799.
2. Kurgan LA, Musilek P. A survey of Knowledge Discovery and Data Mining process models. Knowl Eng Rev. 2006;21:1–24.
3. Mariscal G, Marbán Ó, Fernández C. A survey of data mining and knowledge discovery process models and methodologies. Knowl Eng Rev. 2010;25:137–166.
4. Fayyad U, Piatetsky-Shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data. Commun ACM. 1996;39:27–34.
5. SAS Institute, Inc. (2017, August 20). Introduction to SEMMA. SAS Enterprise Miner 14.3. Available online at http://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm.
6. Wirth R, Hipp J. CRISP-DM: Towards a standard process model for data mining. In: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. London, UK: Springer-Verlag, 2000, Vol. 6, pp. 29–39.
7. Oztekin A, Delen D, Kong ZJ. Predicting the graft survival for heart-lung transplantation patients: An integrated data mining methodology. Int J Med Inform. 2009;78:e84–e96.
8. Topuz K, Zengul FD, Dag A, Almehmi A, Yildirim MB. Predicting graft survival among kidney transplant recipients: A Bayesian decision support model. Decis Support Syst. 2018;106:97–109.
9. Montgomery DC. Design and Analysis of Experiments, 8th ed. Hoboken, NJ: John Wiley & Sons, 2017.
10. Kwak N, Choi C-H. Input feature selection for classification problems. IEEE Trans Neural Netw. 2002;13:143–159.
11. Yang C-H, Huang C-C, Wu K-C, Chang H-Y. A novel GA-Taguchi-based feature selection method. In: Fyfe C, Kim D, Lee S-y, Yin H (Eds.): Intelligent Data Engineering and Automated Learning—IDEAL 2008. Springer Verlag: Berlin Heidelberg, 2008, pp. 112–119.

12. Suzuki A, Ryu K. Feature selection method for estimating systolic blood pressure using the Taguchi method. IEEE Trans Industr Inform. 2014;10: 1077–1085.
13. Chuang L-Y, Yang C-S, Wu K-C, Yang C-H. Correlation-based gene selection and classification using Taguchi-BPSO. Methods Inf Med. 2010;49:254–268.
14. Allias N, Megat MN, Noor M, Ismail MN. A hybrid Gini PSO-SVM feature selection based on Taguchi method: An evaluation on email filtering. In: Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication. New York, NY: ACM, 2014, pp. 1–5.
15. Maji U, Mitra M, Pal S. Imposed target based modification of Taguchi method for feature optimisation with application in arrhythmia beat detection. Expert Syst Appl. 2016;56:268–281.
16. Khaw JFC, Lim BS, Lim LEN. Optimal design of neural networks using the Taguchi method. Neurocomputing. 1995;7:225–245.
17. Wang Q, Stockton DJ, Baguley P. Process cost modelling using neural networks. Int J Prod Res. 2000;38:3811–3821.
18. Ortiz-Rodriguez JM, Martinez-Blanco Mr, Vega-Carrillo Hr. Robust design of artificial neural networks applying the Taguchi methodology and DoE. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA'06). Washington, DC: IEEE, 2006, pp. 131–136.
19. Tortum A, Yayla N, Çelik C, Gökdağ M. The investigation of model selection criteria in artificial neural networks by the Taguchi method. Physica A Stat Mech Appl. 2007;386:446–468.
20. Balestrassi PP, Popova E, Paiva AP, Marangon Lima JW. Design of experiments on neural network's training for nonlinear time series forecasting. Neurocomputing. 2009;72:1160–1178.
21. Hsu W-C, Yu T-Y. E-mail spam filtering based on support vector machines with Taguchi Method for Parameter Selection. J Convergence Inf Technol. 2010;5:78–88.
22. Chan KY, Khadem S, Dillon TS, et al. Selection of significant on-road sensor data for short-term traffic flow forecasting using the Taguchi method. IEEE Trans Industr Inform. 2012;8:255–266.
23. Bashiri M, Geranmayeh AF. Tuning the parameters of an artificial neural network using central composite design and genetic algorithm. Scientia Iranica. 2011;18:1600–1608.
24. Huang ML, Hung YH, Lin ENJU. Effects of SVM parameter optimization based on the parameter design of Taguchi method. Int J Artif Intell Tools. 2011;20:563–575.
25. Lin TY, Ping HC, Hsu TH, et al. A systematic approach to the optimization of artificial neural networks. In: 2011 IEEE 3rd International Conference on Communication Software and Networks. Piscataway, NJ: IEEE, 2011, pp. 76–79.
26. Otok BW, Suhartono, Ulama BSS, Endharta AJ. Design of experiment to optimize the architecture of wavelet neural network for forecasting the tourist arrivals in Indonesia. In: Abd Manaf A, Zeki A, Zamani M, Chuprat S, El-Qawasmeh E (Eds.): Informatics Engineering and Information Science. Berlin, Heidelberg: Springer Verlag, 2011, pp. 14–23.
27. Ardalani-Farsa M, Zolfaghari S. Taguchi's design of experiment in combination selection for a chaotic time series forecasting method using ensemble artificial neural networks. Cybern Syst. 2013;44:351–377.
28. Erfanifard Y, Behnia N, Moosavi V. Tree crown delineation on UltraCam-D aerial imagery with SVM classification technique optimised by Taguchi method in Zagros woodlands. Int J Image Data Fusion. 2014;5:1–15.
29. Huang M-L, Hung Y-H, Lee WM, et al. SVM-RFE based feature selection and Taguchi parameters optimization for multiclass SVM classifier. ScientificWorldJournal. 2014;2014:795624.
30. Kumar D, Gupta AK, Chandna P, Pal M. Optimization of neural network parameters using Grey—Taguchi methodology for manufacturing process applications. Proc Inst Mech Eng Part C. 2015;229:2651–2664.
31. Pontes FJ, Amorim GF, Balestrassi PP, et al. Design of experiments and focused grid search for neural network parameter optimization. Neurocomputing. 2016;186:22–34.
32. Tyasnurita R, Ozcan E, John R. Learning heuristic selection using a Time Delay Neural Network for Open Vehicle Routing. In: 2017 IEEE Congress on Evolutionary Computation (CEC). Piscataway, NJ: IEEE, 2017, pp. 1474–1481.
33. Zare M, Behnia N, Gabriels D. Assessment of land cover changes using Taguchi-based optimized SVM classification approach. J Ind Soc Remote Sens. 2019;47:45–52.
34. Peterson GE, St. Clair DC, Aylward SR, Bond WE. Using Taguchi's method of experimental design to control errors in layered perceptrons. IEEE Trans Neural Netw. 1995;6:949–961.
35. Yang SM, Lee GS. Neural network design by using Taguchi method. J Dyn Syst Meas Control. 1999;121:560–563.
36. Packianather MS, Drake PR, Rowlands H. Optimizing the parameters of multilayered feedforward neural networks through Taguchi design of experiments. Qual Reliab Eng Int. 2000;16:461–473.
37. Kim Y-S, Yum B-J. Robust design of multilayer feedforward neural networks: An experimental approach. Eng Appl Artif Intell. 2004;17: 249–263.
38. Sukthomya W, Tannock J. The optimisation of neural network parameters using Taguchi's design of experiments approach: An application in manufacturing process modelling. Neural Comput Appl. 2005;14:337–344.
39. Garfield E. Historiographic mapping of knowledge domains literature. J Inf Sci Eng. 2004;30:119–145.
40. UNOS. What is UNOS? | About United Network for Organ Sharing. Virginia, USA, 2021.
41. Dag A, Oztekin A, Yucel A, et al. Predicting heart transplantation outcomes through data analytics. Decis Support Syst. 2017;94:42–52.
42. Medved D, Ohlsson M, Höglund P, et al. Improving prediction of heart transplantation outcome using deep learning techniques. Sci Rep. 2018;8:3613.
43. Yoon J, Zame WR, Banerjee A, et al. Personalized survival predictions via Trees of Predictors: An application to cardiac transplantation. PLoS One. 2018;13:e0194985.
44. Dolatsara HA, Chen Y-J, Evans C, et al. A two-stage machine learning framework to predict heart transplantation survival probabilities over time with a monotonic probability constraint. Decis Support Syst. 2020; 137:113363.
45. Kuhn M, Others. Building predictive models in R using the caret package. J Stat Softw. 2008;28:1–26.
46. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281–305.
47. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: with Applications in R. New York, NY: Springer, 2014.
48. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intell Data Anal. 2002;6:429–449.
49. Cohen J. Statistical power analysis for the behavioral analysis, 2nd Ed. Lawrence Erlbaum Associates: New York, 1988.
50. Richardson, John TE. Eta squared and partial eta squared as measures of effect size in educational research. Educ Res Rev. 2011;6:135–147.
51. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python. J Mach Learn Res. 2011;12:2825–2830.
52. Malkomes G, Schaff C, Garnett R. Bayesian optimization for automated model selection. In: Hutter F, Kotthoff L, Vanschoren J (Eds.): Proceedings of the Workshop on Automatic Machine Learning. New York, NY: PMLR, 2016, pp. 41–47.
53. Dag A, Topuz K, Oztekin A, et al. A probabilistic data-driven framework for scoring the preoperative recipient-donor heart transplant survival. Decis Support Syst. 2016;86:1–12.
54. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and Regression Trees. Boca Raton, FL: CRC Press, 1984.
55. Kandanaarachchi S, Muñoz MA, Hyndman RJ, Smith-Miles K. On normalization and algorithm selection for unsupervised outlier detection. Data Min Knowl Discov. 2020;34:309–354.
56. Goos P, Jones B. Optimal Design of Experiments: A Case Study Approach. Oxford, UK: John Wiley & Sons, 2011.
57. Ribeiro MT, Singh S, Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY: Association for Computing Machinery, 2016, pp. 1135–1144.
58. Doshi-Velez F, Kim B. Towards A Rigorous Science of Interpretable Machine Learning. Mach Learn 2017:arXiv [stat.ML].
59. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, et al. (Eds.): Advances in Neural Information Processing Systems 30. Neural Information Processing Systems Foundation, Inc. (NIPS), San Diego, CA, 2017, pp. 4765–4774.
60. Molnar C. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Lulu.com, Morrisville, NC, 2020.
61. Baker M. 1,500 scientists lift the lid on reproducibility. Nature. 2016;533: 452–454.
62. Peng R. The reproducibility crisis in science: A statistical counterattack. Significance. 2015;12:30–32.

63. Hutson M. Artificial intelligence faces reproducibility crisis. Science. 2018; 359:725–726.
64. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–357.
65. Menardi G, Torelli N. Training and assessing classification rules with imbalanced data. Data Min Knowl Discov. 2014;28:92–122.
66. Yu L, Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Fawcet T, Mishra N. (Eds.): Proceedings of the 20th International Conference on Machine Learning (ICML-03), Menlo Park, CA: AAAI Press, 2003, pp. 856–863.
67. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Series B Stat Methodol. 1996;58:267–288.
68. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40:16–28.
69. Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. J Mach Learn Res. 2001;2:97–123.
70. Friedman J, Hastie T, Tibshirani R, et al. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). Ann Stat. 2000;28:337–407.
71. Chen T. xgboost: eXtreme Gradient Boosting. R package version 1.4.1.1. 2021.

Cite this article as: Ahady Dolatsara H, Chen YJ, Leonard RD, Megahed FM, Jones-Farmer LA (2021) Explaining predictive model performance: an experimental study of data preparation and model choice. *Big Data* 3:X, 1–16, DOI: 10.1089/big.2021.0067.

Abbreviations Used

ANN = artificial neural network
AUC = area under the receiver operating characteristic curve
AUPRC = area under the precision recall curve
CRISP-DM = cross-industry standard process for data mining
df = degrees of freedom
DoE = design of experiments
DT = Decision Tree
FFS = Fast Correlation-Based Feature Selection
KDDM = knowledge discovery and data mining
KPLSR = kernel partial least square regression
LASSO = LASSO regression
LDA = linear discriminant analysis
LR = logistic regression
ML = machine learning
NB = naive Bayes
OFAT = one factor at a time
OPTN = Organ Procurement and Transplantation Network
RF = random forest
ROSE = random oversampling examples
SMOTE = Synthetic Minority Oversampling TEchnique
SVM = support vector machines
TMFS = Taguchi method in feature selection
UNOS = United Network for Organ Sharing
XGB = eXtreme Gradient Boosting