Proceedings of the Human Factors and Ergonomics Society Annual Meeting
2023, Vol. 67(1) 2162–2167

DOI: 10.1177/21695067231192258
journals.sagepub.com/home/pro
Sage

Effect of Machine Learning Cross-validation Algorithms Considering Human Participants and Time-series: Application on Biometric Data Obtained from a Virtual Reality Experiment

Ricardo Palma Fraga[1], Ziho Kang[1], and Clare M. Axthelm[1]

Abstract

Data containing human participants and time-series features, as is commonplace in Human Factors research, require special considerations when used in machine learning applications. Ignoring such features during cross-validation procedures might lead to artificially increased model performances due to temporal (i.e. using future observations to predict the present) and participant (i.e using sub-data sets coming from the same participant for training and testing) data leakage. We propose a comparison approach to assess the model performance when machine learning algorithms are trained with two distinctly different cross-validation algorithms: k-fold, which assumes data independence, and population-informed forward chain (PIFC), which accounts for human participants and time-series features. A case study was conducted by using biometric measurements collected from a virtual reality chess experiment. The results show that substantial overestimation might occur when applying the k-fold algorithm instead of the PIFC algorithm.

Keywords

machine learning, virtual reality, cross-validation, eye tracking, fNIRS, data leakage, biometric data

Introduction

Cross-validation is a procedure used to train and evaluate machine learning models by determining how the data needs to be partitioned into training and validation (i.e. testing) data sets, as well as the best-performing hyperparameters that define the model. It is good practice to choose a cross-validation algorithm that represents the problem one is trying to model. Various cross-validation algorithms, such as k-fold (Stone, 1974) and forward chain (or rolling-origin) (Tashman, 2000; Hyndman & Athanasopoulos, 2021) are distinctively different. The k-fold algorithm assumes that observations contained in a data set are independent of each other, whereas the forward chain algorithm assumes that data are dependent on time.

The choice of a cross-validation algorithm might have an effect on model performance (e.g. accuracy), especially when machine learning models are fed biometric data (e.g. eye tracking data, haptic interaction data, cerebral hemodynamic activity data) which can depend on factors such as time and participant. Depending on the application, an improper cross-validation procedure can introduce temporal and/or participant data leakage into the model during training that could lead to performance overestimation. Temporal data leakage may occur when data from future time periods are inadvertently used during the training and/or evaluation (i.e. trying to predict the a present observation using future ones) of a predictive model leading to performance overestimation (Kaufman et al, 2021). Furthermore, participant data leakage may occur when data from the same participant is used for both training and evaluation, which might also lead to performance overestimation (Dehghani et al, 2019). In short, not accounting for participant and time characteristics, when needed, might lead researchers to obtain incorrect or unrealistic predictive model performances.

To address the issues of temporal and participant data leakages, Cochrane and colleagues (2021) proposed an enhanced algorithm called "population-informed forward chain (PIFC) algorithm" which conducts cross-validation considering both users and time elements. Existing research efforts investigate the possible bias on assessing model

[1]University of Oklahoma, Norman, OK, USA

Corresponding Author:
Ricardo Palma Fraga, University of Oklahoma, 202 W. Boyd St., Norman, OK 73019-0390, USA.
Email: rpalmafr@ou.edu

performance (Varma & Simon, 2006), and PIFC algorithm was successfully implemented to analyze the temporal network data of an IBM cloud server (Ohana et al., 2022).

However, to the best of our knowledge, no officially published research exists on the extent of overestimation when k-fold is used instead of population-informed forward chain using actual human experiment data, specifically humans' biometric (or physiological) data.

Prior research by Dehghani et al (2019) compared the effects of a subject-dependent (i.e. k-fold) vs. a subject-independent (a modified k-fold algorithm to account for participant and time data leakage) cross-validation using a human motion data consisting of 17 participants and 33 fitness activities collected over time. They report that subject-dependent cross-validation, which assumes that observations are independent, had higher performances across all their classifiers (ranging from 10% to 21%) compared to subject-independent cross-validation.

Our research goal was to verify whether similar performance overestimation might be obtained when using human experiment data, specifically biometric (or physiological) data collected from multiple participants over time. The differences between our proposed comparison approach (explained in a section below) with those of Dehghani and colleagues (2019) is that (1) we evaluated and compared performance estimations using root mean square deviation (RSME), as our machine learning problem is one of regression and not classification, (2) k-fold and PIFC algorithms, the latter of which came after Dehghani work was published, were applied to commonly used machine leaning algorithms (i.e. multiple linear regression, ridge regression, and random forest), and (3) plausible procedures were defined and developed to compare cross-validation algorithms.

In this research, three machine learning algorithms were considered: multiple linear regression, ridge regression and random forest. Multiple linear regression has been widely for prediction using human experiment data (e.g. Ibrahim & Rusli, 2007). Ridge regression, closely related to multiple linear regression, was considered since it penalizes complex models to prevent overfitting (de Vlaming & Groenen, 2015). Lastly, random forest, unlike the other models, is an ensemble learner, meaning that it creates several decision trees and aggregates their results to produce a single output, which perform better even with relatively smaller data set (Boulesteix et al., 2012).

The application used to obtain biometric data for analysis was a virtual reality chess game. Virtual reality can provide immersive capabilities that can increase user engagement (Bodzin et al, 2021; Allcoat & von Mühlenen, 2018). As part of a larger effort, we were able to establish a multi-person virtual reality system and developed apps that enables the collection of biometric measures such as eye fixations, pupil size, haptic interactions, and cerebral hemodynamic activity data (Kang et. al, under review). Note that a variety of biometric data collected using a low frequency (e.g. 100hz) for a relatively short time from a single participant can generate tens of thousands of data points.

Objective

The objective of the present study was to evaluate the effect of machine learning model performance when time-series data collected from multiple participants are treated as independent vs. dependent (on time and participant). The evaluation is conducted by comparing the RMSEs of three machine leaning algorithms (i.e. linear regression, ridge regression, and random forest) when different cross-validation algorithms (i.e. k-fold vs. population-informed forward chain) are applied. This was investigated through a simplified virtual reality experiment of a chess game.

Proposed Comparison Approach

The proposed comparison approach builds upon the prior research efforts (Dehghani et al., 2019; Cochrane et al., 2021) by introducing detailed steps, procedures, and measures, that were adapted to better compare the model performances when different cross validation algorithms (i.e. k-fold vs. population-informed forward chain (PIFC)) are implemented. To better illustrate the procedures, we assume 6 participants (P1-P6) and 3 time-ordered trials per participant. The assumptions matched with the case study are explained later.

Step 1. Define the parameters and procedures using similar amount of validation data sets to compare the cross-validation algorithms

Each cross-validation procedure can be visualized in **Figure 1**. Regarding the k-fold cross-validation, firstly, all data are randomly split into training and test sets. The training data contains 80% of the data and the remaining 20% are allocated for testing. Secondly, the training data are again split into k-folds (i.e. subsets), and a fold (i.e. subset) is used to validate the model. Note that the test data set is different from the validation data set and is used to assess model performance on unseen data once the cross-validation procedures identifies the best performing models.

In the case of the PIFC, firstly, data are split based on participants: a train set (i.e. P2-P6) and a test set (i.e. P1). Note that the percentage of data used for training can slightly differ. In the below example, P2-P6 contains approx. 83% and P1 contains approx. 17%. Secondly, each validation fold is used against the data available until the point-in-time (see pink area in **Figure 1(b)**). For example, when P6 is used for validation, 3 folds are created (since we assumed 3 time-ordered trials) using the time-ordered data of P6. When using the first fold of P6, the remaining data are not included to prevent data leakage from future samples (see grey area in **Figure 1(b)**). Three validation folds are used per participant;

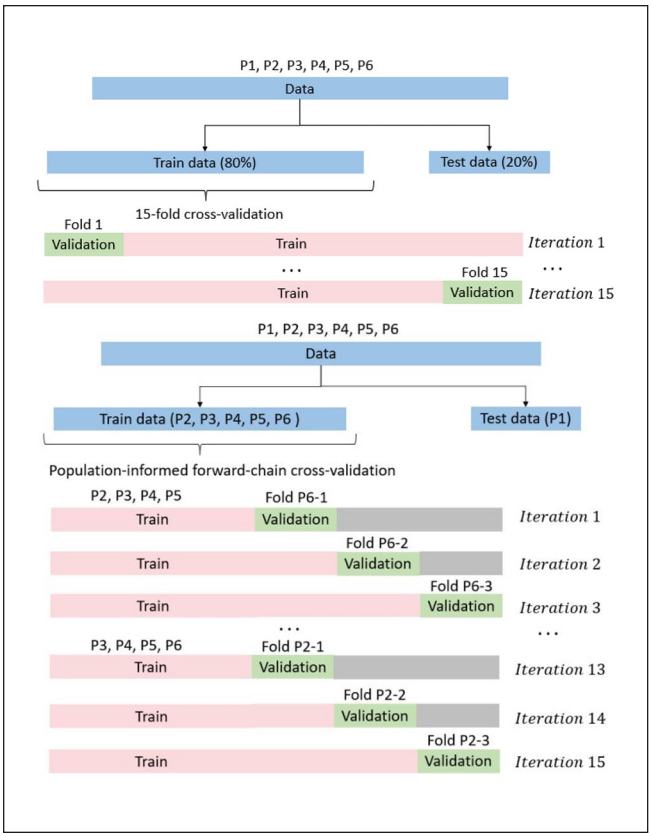

Figure 1. Cross validation algorithms: In each cross-validation fold, the light pink denotes the data used for training, while the light green represents the data used for testing. Note that the PIFC process explained by Cochrane and colleagues (2021) is visualized here. In our case, each fold results in an RMSE value. For the population-informed (right), the grey color indicates data that are assumed to be not temporarily available at that fold.
(a) K-fold cross-validation: Example using 15-fold.
(b) Population-informed forward chain cross-validation (PIFC).

therefore, a total of fifteen folds are used (resulting in fifteen iterations) when we consider all five participants.

Step 2. Apply each cross-validation algorithm to different machine learning algorithms to generate combinations of prediction models

Many machine learning algorithms can be used for training and testing, and we considered three popular machine learning algorithms: multiple linear regression, ridge regression, and random forest. For both cross-validation algorithms, each iteration produces the RMSE when a given machine learning algorithm is applied. For example, if we have two cross-validating algorithms and three machine learning algorithms, then a total of six combinations can exist: {(k-fold, linear regression), (k-fold, ridge regression), (k-fold, random forest), (population-informed forward chain, linear regression), (population-informed forward chain, ridge regression), and (population-informed forward chain, random forest)}.

Step 3. Obtain the RMSE values

RMSE values are obtained for each iteration. Using the assumptions provided above, in the case of k-fold cross-validation, the fifteen folds are used; therefore, fifteen iterations are performed which generates fifteen RMSE values. Similarly, for PIFC, fifteen RMSE values are obtained due to the training data containing five participants and each participant having three time-ordered trials.

Step 4. Obtain the magnitude of overestimation by comparing the average RMSE values among the prediction models

The average RSME values are used to calculate performance differences through equation (1) shown below. In (1), *PD* represents the performance difference (as a percentage), x_{kfold} is the performance of the k-fold cross-validation trained model (in RMSE), and x_{PIFC} is the performance of the population-informed forward chain trained model (in RMSE).

$$PD = \frac{x_{kfold} - x_{PIFC}}{x_{PIFC}} \times 100 \quad (1)$$

Case Study

A case study was conducted through an experiment of a fully immersive virtual reality chess game environment. The six combinations (explained in Step 2 within the Proposed Comparison Approach section) are used to evaluate the magnitude of overestimations among the combinations. The results might help us to better understand the magnitude of the overestimation effect before the predictive model is chosen.

Participants and Data Points: Eight participants were recruited from the University of Oklahoma. Ages ranged between 20 and 40, and participants were identified as non-expert chess players who play chess casually. Due to the low quality of the oxygenation/de-oxygenation data collected, two participants' data were not included in the analysis. The data collected consisted of a matrix composed of 3965 rows and 16 features, totaling 63,440 data points, including both dependent and independent variables, among 6 participants.

Apparatus: HTC Vive Pro Eye Virtual Reality (VR) device was used to collect the eye and hand movements. The VR device is equipped with an eye tracking system running at 120 Hz sampling rate having a spatial accuracy of 0.5° - 1.1°. Vive Pro controllers were used to interact within VR. fNIR Model 2000C (BIOPAC Systems, Goleta, CA, USA)

Figure 2. A participant's eye movements at the end of a scenario. The yellow dots represent the location of gazes made by the participant, and the red lines constitute the saccades in-between said gazes.

was used to collect oxygenation data (Ayaz, 2005). Vizard Virtual Reality API functions (version 7.1; WorldViz, Santa Barbara, CA, USA) were used to synchronize all data.

Scenarios: Human-sized chess pieces were used to create an enjoyable and engaging virtual reality (VR) environment. Three scenarios were re-created based on the existing famous chess games played available at Chess.com (e.g. Kasparov vs. Topalov, Aronian vs. Anand). Each scenario was set up so that the participant had to move a chess piece based on how the opponent (expert) moved a chess piece. The opponent's move was chosen through the suggested move generated in Chessbase.com. An example of the VR environment created along with captured eye movements from a participant is provided in Figure 2.

Task and Procedure: Prior to the experiment, participants were trained on how to navigate through the VR space and move chess pieces. They were instructed to move a chess piece after the opponent moved a chess piece. The opponent's action was conducted by the researcher. Participants were instructed to make a move (using white pieces) that would move them one step closer towards winning the game, which would be followed by a move by the researcher (using black pieces). The scenario would end after they had completed a total of three moves, after which another scenario would be loaded until the participant completed all three scenarios (i.e. three trials).

Measures: For each scenario, eye movement, hand movements, and oxygenation/de-oxygenation data of the participants were collected. In detail, for the eye movement data, the variables consist of the location x, y, and z of the gaze in the virtual reality environment (point x, point y, and point z), the cumulative time until the next gaze on a chess piece (tTFix), as well as the pupil diameter (pupil diameter). In terms of hand movements, the location x, y, and z of both the left and right hands, relative to the body of the user, were included (lhand x, lhand y, and lhand z for the left hand; rhand x, rhand y, and rhand z for the right hand). Lastly, several variables were feature engineered, such as the distance from the gaze point to the left and right hands (distGLH for the left hand and distGRH for the right hand), as well as the distance between the left and right hands (distRLH). Finally, for the brain activity data, oxygenation and de-oxygenation data were collected, then the difference between the oxygenation value and deoxygenation value was calculated (oxyDeoxyDiff), which indicates the brain activity level.

Data analysis: The models were trained through two cross-validation methods: 15-fold cross-validation and a population-informed forward chain cross-validation (as described in Cochrane et al, 2021). Differences in performance on the train and test data sets, quantified through root mean squared error (RMSE), are reported. Prior to training, as the data contains different units, the features were scaled to a range between 0 and 1, based on the observed maximum and minimum values in their corresponding train sets. Three machine learning algorithms, multiple linear regression, ridge regression, and random forest, were implemented and trained using the two cross-validation approaches. During both cross-validation procedures, a wide range of hyperparameter values were explored. For ridge regression, alpha values of .001, .01, .05, .1, .5, 1, 5, 10, 50, and 100 were considered, which penalizes overly complex models that can lead to overfitting. For the random forest model, two hyperparameters were explored to control overfitting: the number of estimators used (10, 50, 100, 500) and the maximum depth of each estimator (3, 6, 9, and 12). Finally, a Pearson correlation matrix was computed to explore the linear relationship between the dependent and independent variables.

Results

The model performance results are provided in Table 1. K-fold cross-validation algorithm shows better performance (i.e., lower averaged RMSE values) compared to the PIFC algorithm. Hyperparameters values obtained from the best performing ridge regression and random forest models are provided in Table 2.

The model performance of the best models on the test data can be seen in Table 3. The performance differences (as percentages) between the k-fold cross-validation and population-informed forward chain cross-validation across the machine learning models and the train/test data sets can be found in Table 4. The k-fold cross-validation trained models report higher performances (i.e. lower RMSE) across all models on the train data, ranging from the lowest increase at 18.58% for ridge regression to the highest increase at 60.67% for random forest. The increased performance trends can also be seen when using the test data, with the exception of the ridge and linear regression models, which saw a slightly higher performance than the PIFC.

The Pearson correlation matrix between the dependent and independent variables is visualized in Figure 3. Note that no dependent variables have a strong positive or negative relationship with the independent variable.

Table 1. Train data: Average model performance (RMSE) across the two cross-validation methods on their respective training data.

Cross-validation method	Multiple Linear Regression	Ridge Regression	Random Forest
K-fold	2.44	2.44	1.17
PIFC	3.42	2.99	2.99

Table 2. Hyperparameter values of the best ridge regression & random forest models between the two cross-validation algorithms.

Cross-validation method	Ridge Regression	Random Forest
K-fold	$\alpha = 1$	500 estimators, max depth of 12
PIFC	$\alpha = 150$	50 estimators, max depth of 9

Table 3. Test data: Average model performance (RMSE) across the two cross-validation methods on their respective test data.

Cross-validation method	Multiple Linear Regression	Ridge Regression	Random Forest
K-fold	2.53	2.53	1.11
PIFC	2.48	2.31	2.22

Table 4. Performance differences (in percentages): Negative percentage indicates a higher performance (i.e. lower RMSE) of the k-fold cross-validation when compared to that of population-informed forward chain cross-validation.

Data set	Multiple Linear Regression	Ridge Regression	Random Forest
Train	-28.70% (2.44 vs 3.42)	-18.58% (2.44 vs 2.99)	-60.67% (1.17 vs 2.99)
Test	+1.85% (2.53 vs 2.48)	+9.35% (2.53 vs 2.31)	-50.09% (1.11 vs 2.22)

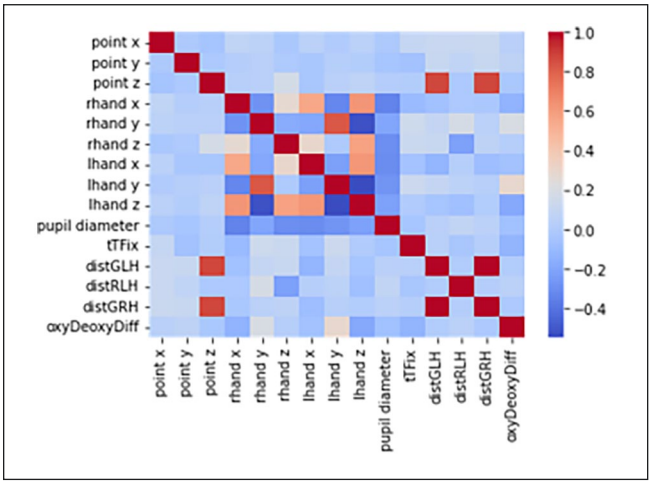

Figure 3. Visualization of the Pearson correlation matrix. Red indicates a positive correlation, while dark blue a negative one.

Discussion, Limitations, and Future Research

Our results indicate a sharp contrast in the performance of the chosen machine learning algorithms depending upon the cross-validation algorithms implemented. Similar to other closely related research (e.g. Dehghani et al, 2019), we found that the models trained under a k-fold cross-validation might provide artificially increased performance results. The best performing model, the random forest model, had a 60.67% lower RMSE on the train data and a 50.09% lower RMSE on the test data when using the k-fold cross-validation procedure, relative to population-informed forward chain cross-validation.

Regarding the k-fold cross-validation algorithm, the random forest algorithm performed substantially better (1.11 RMSE in the test data) than the other two models (2.53 RMSE in the test data for both). One reason behind this may be that the random forest algorithm can handle non-linear relationships between the dependent and independent variables better than the linear algorithms. However, the clear model choice highlighted by the k-fold cross-validation procedure cannot be seen in the more appropriate population-informed forward chain cross-validation algorithm.

Therefore, based on the results of the study, one may inappropriately conclude that the random forest model may be the most suitable algorithm to predict engagement when using a relatively small set of data among a limited number of participants. However, when accounting for the temporal and subject

aspects contained in the data, as was done when implementing the PIFC algorithm, none of the machine learning algorithms chosen substantially differed in performance.

In addition, the Pearson correlation matrix shows that the input features used might not be capable of effectively predicting hemodynamic brain activities. It may be reasonable to believe that the results shown in the matrix somewhat better accord with the results obtained from the PIFC algorithm. In other words, it is plausible that these particular set of features, when used as inputs in a machine learning algorithm, should not generate an accurate model in the first place, particularly in the linear models.

Lastly, we believe, and agree on the idea that we want to train and tune the model in a way that reflects the application and environment it will be deployed in, specially to "simulate the real-world forecasting environment" (Tashman, 2000). In our case, PIFC served as an appropriate cross-validation procedure that represented our application. On the other hand, for example, if a group of researchers were interested in predicting the state (i.e. cognitive engagement or any other measure) of a user with every hour for an experimental or a real-life application, then they should train and assess the performance of their algorithms to reflect such an application.

Future research involves better defining the measures to predict the cerebral hemodynamic activities, such as using eye fixations and durations on the areas (or targets) of interests, visual scanning patterns and hand movement patterns, interaction behaviors, among others. In addition, more data should be collected, and experiment design can be improved to better isolate incidents that lead to substantially increased brain activities. Finally, other machine learning algorithms will be considered that could better predict user engagement (i.e. brain activities) in virtual reality. For example, support vector machines, and their ability to incorporate several distinct kernels, could have more predictive capabilities than our linear regression and regularized models.

Acknowledgments

This material was based upon work supported by the National Science Foundation under Grant No. 1943526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ORCID iD

Ziho Kang https://orcid.org/0000-0003-4058-4584

References

Allcoat, D., & von Mühlenen, A. (2018). Learning in virtual reality: Effects on performance, emotion and engagement. *Research in Learning Technology*, *26*.

Ayaz, H. (2005). *Analytical software and stimulus-presentation platform to utilize, visualize and analyze near-infrared spectroscopy measures*. [MS Thesis, Drexel University].

Bodzin, A., Junior, R. A., Hammond, T., & Anastasio, D. (2021). Investigating engagement and flow with a placed-based immersive virtual reality game. *Journal of science education and technology*, *30*, 347-360.

Boulesteix, A. L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(6), 493-507.

Cochrane, C., Ba, D., Klerman, E. B., & Hilaire, M. A. S. (2021). An ensemble mixed effects model of sleep loss and performance. *Journal of theoretical biology*, *509*, Article 110497.

Dehghani, A., Sarbishei, O., Glatard, T., & Shihab, E. (2019). A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors. *Sensors*, *19*(22), 5026.

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia.

Ibrahim, Z., & Rusli, D. (2007). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In *21st Annual SAS Malaysia Forum, Malaysia*.

Kang, Z, Palma Fraga, R, Izzoteglu, K, Lee, J, Deering, D. D, & Arana, W. X. (under review). Development of a smart learning application in multi-person virtual reality using biometric measures of neuroimaging, eye tracking, and haptic interactions, In *Proceedings of the 67th Meeting of the Human Factors and Ergonomics Society*. Washington D.C.

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(4), 1-21.

Sabri, H., Cowan, B., Kapralos, B., Porte, M., Backstein, D., & Dubrowskie, A. (2010). Serious games for knee replacement surgery procedure education and training. *Procedia-social and behavioral sciences*, *2*(2), 3483-3488.

Ohana, D., Wassermann, B., Dupuis, N., Kolodner, E., Raichstein, E., & Malka, M. (2022). Hybrid anomaly detection and prioritization for network logs at cloud scale. *Proceedings of the Seventeenth European Conference on Computer Systems, France*, 236-250.

Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111–147, 1974.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, *16*(4), 437-450.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(1), 1-8.

de Vlaming, R., & Groenen, P. J. (2015). The current and future use of ridge regression for prediction in quantitative genetics. *BioMed research international*, Article 143712.