**Supplement S1**

Table of Contents

[1. Calculating the score in R 2](#_Toc48890535)

[2. Github repository 3](#_Toc48890536)

[3. Table S1: EPIC data CpGs: Model Quality measures across Raine Study, NFBC1986 and NFBC1966 4](#_Toc48890537)

[4. Figure S1: Distribution of smoking variable and smoking score over plates in the Raine Study 5](#_Toc48890538)

[5. Figure S2: Model quality measures for all machine learning models 6](#_Toc48890539)

# 1. Calculating the score in R

The elastic net regression weights that were extracted from the final model and that are reported in the Supplement Excel tables S and SX for 450k and EPIC are implemented in out R package DNAsmokeR. The following contains step by step instructions as to how to create the score based on DNA methylation data.

1. ***Install the DNAsmokeR R package***

The R code to apply the score can be found on github:

<https://github.com/Hobbeist/DNAsmokeR>.

This is a package that can be installed in R using devtools::install\_github(“DNAsmokeR”, build\_vignettes=TRUE).

After installing the package, the function smokeScore() will be available, once the package is loaded via “library(DNAsmokeR)”.

The following are the steps one needs to follow to create the score:

1. ***Loading your DNA methylation data***

Reading the data:

Your DNA methylation data needs to be in the following format:

columns = CpGs,

rows = Study participants.

CpG <- readr::read\_csv(path/to/your/DNA-methylation/data.csv”)

1. ***Using the smokeScore function***

smokeScore(data, ARRAY, class)

where:

* data is your CpG data set (CpG),
* ARRAY is one of the following (including quotation marks):
	+ “450K”
	+ “EPIC”
* Class, specifying if you want to return a class or a probability score:
	+ “class” for class
	+ “prob” for probability score.

# 2. Github repository

All code for the analysis of this manuscript can be found on github:

<https://github.com/Hobbeist/Supplement_Code_Smoking_Score>

The repository contains the following:

**models**

This folder contains the trained caret model objects for:

* Gradient Boosting Machine (gradientBoosting\_model.rds)
* Random Forest (randomForest\_model.rds)
* Support Vector Machine (supportVector\_model.rds)

Those can be downloaded and loaded into an R session via the command readRDS(“location/of/the/model/file.rds”)

**r-code**

This folder contains the following code:

* Training the elastic net regression model: 1-training-elastic-net-model.R
* Training and tuning the Gradient Boosting Machine, Random Forest and Support Vector Machine models: 2-Tuning-GBM-RF-SVM-Models.R
* Creating the Reese and Richmond Scores: 3-Reese-and-Richmond-scores.R
* Using the Gradient Boosting Machine, Random Forest and Support Vector Machine objects in the model folder to create a smoking score based on those models: 4-Predict-rf-gbm-svm.R

**Of note**: The repository is self-containing. This means, when the full repository is downloaded, the folder contains the file Supplement\_2.Rproj. Given R and RStudio are installed, double clicking this file will launch a RStudio session and the Predict-rf-gbm-svm.R code will load the caret models for the respective models without changing the directory location.

# 3. Table S1: EPIC data CpGs: Model Quality measures across Raine Study, NFBC1986 and NFBC1966

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|   | **Sensitivity** | **Specificity** | **Cohen’s kappa** | **Accuracy** | **AUC** | **Brier Score** | **# CpGs required** |
| **Raine Study test data set** |  |  |  |  |  |  |  |
| Elastic Net Score | 0.85 | 0.83 | 0.65 | 0.84 | 0.905 | 0.16 | 181 |
| **NFBC1986** |  |  |  |  |  |  |  |
| Elastic Net Score | 0.77 | 0.76 | 0.43 | 0.77 | 0.829 | 0.17 | 181 |
| **NFBC1966** |  |  |  |  |  |  |  |
| Elastic Net Score | 0.76 | 0.62 | 0.33 | 0.77 | 0.768 | 0.19 | 181 |
|  |  |  |  |  |  |  |  |

# 4. Figure S1: Distribution of smoking variable and smoking score over plates in the Raine Study



# 5. Figure S2: Model quality measures for all machine learning models

**Figure S1**. Accuracy and Cohen’s kappa for all tested models. Both statistics were dervived during the cross-validation step of the model training. The data is split into random training and test sets during training and the metrics are calculated based on the test set. The average and the 95% confidence intervals are provided for every model. Both measures are unit-less and the higher the value, the better the performance for both accuracy and Cohen’s .

glmnet: Elastic net regression

gbm: Gradient Boosting Machine

rf: Random Forest

svmLinear: Support Vector Machine with Linear Kernel

c50: C5.0

lda: Lineat Discriminant Analysis

bagging: Tree Bagging Algorithm

cart: Classification and Regression Tree

nb: Naïve Bayes

logisit: Logistic Regression

knn: k-Nearest Neighbour

