Prediction Model

Beyza Hizli, Nele Albers

22 September, 2022

Contents

Introduction

This file is to show how we arrived at our final prediction model for how motivating the examples from other people are perceived. This includes reproducing Table 1 from the paper.

Authored by Nele Albers, Beyza Hizli, Bouke L. Scheltinga, Eline Meijer, and Willem-Paul Brinkman.

Required and output files

These files are required: motivation_final.csv and similarity_predictions.csv.

And these files are created: Figures/km_clustered.png, clusters.csv, and centers.csv.

Setup

Let's load the packages we need.

```
library(factoextra)
library(formatR)
library(ggplot2)
library(lme4)
```

```
library(lm.beta)
library(MASS)  # for stepwise regression
```

Load data

And we load the necessary data. If you want to re-create this data file, use the notebook "preprocess_for_model.ipynb."

```
# Read data
mot_ratings <- read.csv(file = 'motivation_final.csv', sep=",")

# Transform data to dataframe
mot_ratings <- as.data.frame(mot_ratings)
```

Model to predict motivation ratings using all independent user variables

First, we fit a model that predicts motivation ratings based on all independent user variables.

```
mot_model <- lm(rating ~ age + education + household_income + personal_income +
    household_size + gender + smoking_freq + socioeconomic_status + weekly_exercise +
    extraversion + agreeableness + conscientiousness + emotional_stability +
    openness_to_experiences + need_for_cognition + ttm_pa + exercise_identity +
    exercise_se + sitting_weekday + sitting_weekend + godin_activity +
    smoke, data = mot_ratings)

summary(mot_model)
```

```
##
## Call:
## lm(formula = rating ~ age + education + household_income + personal_income +
##     household_size + gender + smoking_freq + socioeconomic_status +
##     weekly_exercise + extraversion + agreeableness + conscientiousness +
##     emotional_stability + openness_to_experiences + need_for_cognition +
##     ttm_pa + exercise_identity + exercise_se + sitting_weekday +
##     sitting_weekend + godin_activity + smoke, data = mot_ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4164 -1.1692  0.2754  1.3085  4.3951
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.77088    0.34435   5.143 3.63e-07 ***
## age                      -0.85353    0.29624  -2.881  0.00410 **
## education                 0.12984    0.32247   0.403  0.68735
## household_income          0.54429    0.46141   1.180  0.23859
## personal_income           0.06821    0.52629   0.130  0.89692
## household_size            0.53056    0.34259   1.549  0.12197
## gender                   -0.05661    0.14139  -0.400  0.68902
## smoking_freq              0.10604    0.33567   0.316  0.75218
## socioeconomic_status     -0.04114    0.36758  -0.112  0.91092
## weekly_exercise           0.10985    0.19595   0.561  0.57527
```

```
## extraversion            -1.01772    0.33936  -2.999  0.00282 **
## agreeableness            0.53826    0.34635   1.554  0.12067
## conscientiousness        0.18920    0.39100   0.484  0.62863
## emotional_stability     -0.07492    0.36848  -0.203  0.83895
## openness_to_experiences -0.95513    0.32226  -2.964  0.00315 **
## need_for_cognition       0.15146    0.33557   0.451  0.65188
## ttm_pa                  -0.70805    0.28395  -2.494  0.01290 *
## exercise_identity       -0.69280    0.41693  -1.662  0.09708 .
## exercise_se             -0.96803    0.31841  -3.040  0.00246 **
## sitting_weekday          0.47081    0.44186   1.066  0.28705
## sitting_weekend          0.60907    0.40373   1.509  0.13190
## godin_activity          -0.97358    0.18453  -5.276 1.82e-07 ***
## smoke                    0.26995    0.15001   1.799  0.07243 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.776 on 625 degrees of freedom
## Multiple R-squared:  0.1699, Adjusted R-squared:  0.1407
## F-statistic: 5.814 on 22 and 625 DF,  p-value: 3.691e-15
```

Similarity ratings for clusters of example people

To potentially improve the prediction model, we wanted to add as independent variables the similarity ratings for clusters of example people. For this, we created clusters for the example people based on the ratings by those 36 people that rated the examples. As not all people rated all example people on similarity, missing values were filled in based item-based collaborative filtering (Sarwar et al. (2001)). Specifically, missing values were predicted based on similarity ratings given to other examples. If you want to reproduce our collaborative filtering approach, go to the file "collaborative_filtering.ipynb."

We then used k-means clustering. This led to 3 clusters since more clusters both resulted in too much overlap between the clusters and would have forced us to use too many of our examples as prototypes to be rated by the users of the goal-setting dialog.

```
sim_matrix <- read.csv(file = "similarity_predictions.csv", sep = ",")
sim_matrix <- t(sim_matrix[, -1])

# output to be present as PNG file
png(file = "Figures/km_clustered.png")
km <- kmeans(sim_matrix, centers = 3, nstart = 100)

# Visualize the clusters, see km_clustered.png image file in the
# folder 'Figures'
fviz_cluster(km, data = sim_matrix)
# saving the file
dev.off()
```

```
## pdf
##   2
```

```
clusters = data.frame(km$cluster)

# Saving clusters en centers to csv
write.csv(clusters, "clusters.csv", row.names = FALSE)
write.csv(km$centers, "centers.csv", row.names = FALSE)
```

Stepwise regression

We now obtained three new independent variables. These were the average ratings per cluster based on the two most centered examples per cluster. Users of the goal-setting dialog also rated these six prototypes on similarity before the dialog. The six prototype examples were thus excluded from the candidate set of examples to be shown in the goal-setting dialog.

Next, we used stepwise regression to find the subset of variables with the lowest prediction error. Stepwise regression uses a combination of forward and backward selection.

```
# We now have 3 new independent variables: c1, c2 and c3
mot_model_plus_sim <- lm(rating ~ age + education + household_income +
    personal_income + household_size + gender + smoking_freq + socioeconomic_status +
    weekly_exercise + extraversion + agreeableness + conscientiousness +
    emotional_stability + openness_to_experiences + need_for_cognition +
    ttm_pa + exercise_identity + exercise_se + sitting_weekday + sitting_weekend +
    godin_activity + smoke + c1 + c2 + c3, data = mot_ratings)

# Run stepwise regression
step_mot_model_plus_sim <- stepAIC(mot_model_plus_sim, direction = "both",
    trace = FALSE)
summary(step_mot_model_plus_sim)
```

```
##
## Call:
## lm(formula = rating ~ age + household_income + household_size +
##     extraversion + openness_to_experiences + ttm_pa + exercise_se +
##     sitting_weekend + godin_activity + c1 + c3, data = mot_ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1095 -1.2212  0.1994  1.1829  4.1854
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.05849    0.37475   0.156  0.87602
## age                       -0.65668    0.27863  -2.357  0.01873 *
## household_income           0.69589    0.35854   1.941  0.05271 .
## household_size             0.55114    0.32439   1.699  0.08981 .
## extraversion              -0.75341    0.32355  -2.329  0.02020 *
## openness_to_experiences   -0.79136    0.30525  -2.593  0.00975 **
## ttm_pa                    -0.63835    0.25130  -2.540  0.01131 *
## exercise_se               -0.82711    0.30075  -2.750  0.00613 **
## sitting_weekend            0.55334    0.36253   1.526  0.12743
## godin_activity            -0.90406    0.17465  -5.177 3.04e-07 ***
## c1                         0.13737    0.05134   2.676  0.00765 **
## c3                         0.39545    0.05794   6.825 2.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.703 on 636 degrees of freedom
## Multiple R-squared:  0.2234, Adjusted R-squared:   0.21
## F-statistic: 16.64 on 11 and 636 DF,  p-value: < 2.2e-16
```

Correlation analysis

Lastly, we wanted to add to the model all those user variables that were significantly correlated with the dependent variable. For this, we conducted a correlation analysis for the user variables that were not already included after the stepwise regression.

```
cor.test(mot_ratings$rating, mot_ratings$education)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$education
## t = -0.11373, df = 646, p-value = 0.9095
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08146701  0.07257125
## sample estimates:
##          cor
## -0.004474425
```

```
cor.test(mot_ratings$rating, mot_ratings$personal_income)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$personal_income
## t = 0.44756, df = 646, p-value = 0.6546
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05949511  0.09449875
## sample estimates:
##        cor
## 0.01760623
```

```
cor.test(mot_ratings$rating, mot_ratings$gender)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$gender
## t = -0.49185, df = 646, p-value = 0.623
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09622529  0.05775871
## sample estimates:
##         cor
## -0.01934802
```

```
cor.test(mot_ratings$rating, mot_ratings$smoking_freq)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$smoking_freq
## t = 0.7628, df = 646, p-value = 0.4459
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.04713112  0.10677240
## sample estimates:
##        cor
## 0.02999843
```

```
cor.test(mot_ratings$rating, mot_ratings$socioeconomic_status)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$socioeconomic_status
## t = -0.50173, df = 646, p-value = 0.616
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09661033  0.05737135
## sample estimates:
##         cor
## -0.01973653
```

```
cor.test(mot_ratings$rating, mot_ratings$weekly_exercise)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$weekly_exercise
## t = -0.66436, df = 646, p-value = 0.5067
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.10294336  0.05099341
## sample estimates:
##         cor
## -0.02612988
```

```
cor.test(mot_ratings$rating, mot_ratings$agreeableness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$agreeableness
## t = 2.2261, df = 646, p-value = 0.02635
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01029788 0.16317348
## sample estimates:
##        cor
## 0.08724934
```

```
cor.test(mot_ratings$rating, mot_ratings$conscientiousness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$conscientiousness
## t = -0.38415, df = 646, p-value = 0.701
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09202595  0.06198040
```

```
## sample estimates:
##        cor
## -0.0151124
```

```
cor.test(mot_ratings$rating, mot_ratings$emotional_stability)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$emotional_stability
## t = 0.52599, df = 646, p-value = 0.5991
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05642004  0.09755573
## sample estimates:
##        cor
## 0.02069053
```

```
cor.test(mot_ratings$rating, mot_ratings$need_for_cognition)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$need_for_cognition
## t = 0.10932, df = 646, p-value = 0.913
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07274386  0.08129464
## sample estimates:
##         cor
## 0.004300904
```

```
cor.test(mot_ratings$rating, mot_ratings$exercise_identity)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$exercise_identity
## t = -3.335, df = 646, p-value = 0.0009018
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.20506504 -0.05361579
## sample estimates:
##        cor
## -0.1300992
```

```
cor.test(mot_ratings$rating, mot_ratings$sitting_weekday)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$sitting_weekday
## t = 0.73994, df = 646, p-value = 0.4596
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04802801  0.10588364
## sample estimates:
```

```
##       cor
## 0.0291003
```

```
cor.test(mot_ratings$rating, mot_ratings$smoke)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mot_ratings$rating and mot_ratings$smoke
## t = 2.2104, df = 646, p-value = 0.02743
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.009682897 0.162574747
## sample estimates:
##        cor
## 0.08663894
```

Final model

Based on the correlation analysis, we included the user variables that had at least a significant small correlation based on (Cohen (1992)) with the dependent variable. Thus, physical activity identity (called "exercise_identity" in the code) was included. Furthermore, we added the variable "c2" that captures the similarity rating for the second cluster.

Now we show the final model we obtained. This reproduces values from Table 1 from the paper.

```
final_model = lm(formula = rating ~ age + household_income + household_size +
    extraversion + openness_to_experiences + ttm_pa + exercise_identity +
    exercise_se + sitting_weekend + godin_activity + c1 + c2 + c3, data = mot_ratings)

summary(final_model)
```

```
##
## Call:
## lm(formula = rating ~ age + household_income + household_size +
##     extraversion + openness_to_experiences + ttm_pa + exercise_identity +
##     exercise_se + sitting_weekend + godin_activity + c1 + c2 +
##     c3, data = mot_ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1366 -1.2280  0.2267  1.1873  4.2668
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.13887    0.39643   0.350  0.72623
## age                       -0.64921    0.27886  -2.328  0.02022 *
## household_income           0.76815    0.36463   2.107  0.03554 *
## household_size             0.60581    0.32879   1.843  0.06586 .
## extraversion              -0.73115    0.32571  -2.245  0.02513 *
## openness_to_experiences   -0.78798    0.30565  -2.578  0.01016 *
## ttm_pa                    -0.53480    0.26768  -1.998  0.04616 *
## exercise_identity         -0.44290    0.39280  -1.128  0.25994
## exercise_se               -0.82005    0.30114  -2.723  0.00664 **
## sitting_weekend            0.56707    0.36612   1.549  0.12191
```

```
## godin_activity          -0.88615    0.17603  -5.034 6.26e-07 ***
## c1                       0.12762    0.05243   2.434  0.01519 *
## c2                      -0.01020    0.05123  -0.199  0.84228
## c3                       0.40091    0.05901   6.794 2.51e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.704 on 634 degrees of freedom
## Multiple R-squared:  0.225,  Adjusted R-squared:  0.2091
## F-statistic: 14.16 on 13 and 634 DF,  p-value: < 2.2e-16
```

And here are the standardized coefficients:

```
lm.beta(final_model)
```

```
##
## Call:
## lm(formula = rating ~ age + household_income + household_size +
##     extraversion + openness_to_experiences + ttm_pa + exercise_identity +
##     exercise_se + sitting_weekend + godin_activity + c1 + c2 +
##     c3, data = mot_ratings)
##
## Standardized Coefficients::
##             (Intercept)                      age        household_income
##             0.000000000            -0.082310751              0.076627862
##          household_size            extraversion openness_to_experiences
##             0.066713676            -0.080603503             -0.093321332
##                  ttm_pa       exercise_identity              exercise_se
##            -0.080875892             -0.045109694             -0.101615882
##         sitting_weekend          godin_activity                       c1
##             0.058593372            -0.197543027              0.089666035
##                      c2                       c3
##            -0.007235487             0.252474906
```

References

Cohen, Jacob. 1992. "A Power Primer." *Psychological Bulletin* 112 (1): 155.

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms." In *Proceedings of the 10th International Conference on World Wide Web*, 285–95.