Data Analytics I: Predictive Econometrics
Assignment II

Lauritz Storch
Sebastian Maser
Ivan Medvedev
Jan Gutjahr

2022-12-01

Data Preparation and Package Installation

Download the data sets student-mat-train.Rdata and student-mat-test.Rdata[1] from Canvas. Load the data into R. Install and load the packages glmnet and corrplot.

```
# # Clear R
# rm(list=ls())
# cat("\014")

## Set working directory to load data
setwd("~/Downloads")


## Packages and Library
# Packages function
install_if_missing <- function(p) {
  if (p %in% rownames(installed.packages()) == F) {
    try(install.packages(p, dependencies = T, type="binary"))
  } else {
    cat(paste("Skipping already installed package:", p, "\n"))
  }
}

# Define packages
packages = c("glmnet","corrplot")

# Install
invisible(sapply(packages, install_if_missing))
```

```
## Skipping already installed package: glmnet
## Skipping already installed package: corrplot
```

```
# read library
for(pkg in packages){
  library(pkg, character.only = TRUE)
}
```

```
## Lade nötiges Paket: Matrix
```

[1] Please make sure your device is connected to the internet and you are logged into Canvas to download the dataset.

```
## Loaded glmnet 4.1-6
```

```
## corrplot 0.92 loaded
```

```r
## Read in data
load("student-mat-train.Rdata")
load("student-mat-test.Rdata")
```

Task 1

Q: How many observations are in the training and test data? (0.5 points)

```r
cat("Number of observations in the training data: ",sum(!is.na(train)))
```

```
## Number of observations in the training data:  5564
```

```r
cat("Number of observations in the test data: ",sum(!is.na(test)))
```

```
## Number of observations in the test data:  3718
```

Task 2

Q: What is the average, minimum, and maximum grade in the training data? (0.5 points)

```r
cat("The average, minimum, and maximum grade in the training data: \n",
    round(summary(train$G3)[["Mean"]],4), " (average), ", summary(train$G3)[["Min."]],
    " (Min), ", summary(train$G3)[["Max."]], " (Max).")
```

```
## The average, minimum, and maximum grade in the training data:
##  11.6402  (average),  4  (Min),  19  (Max).
```

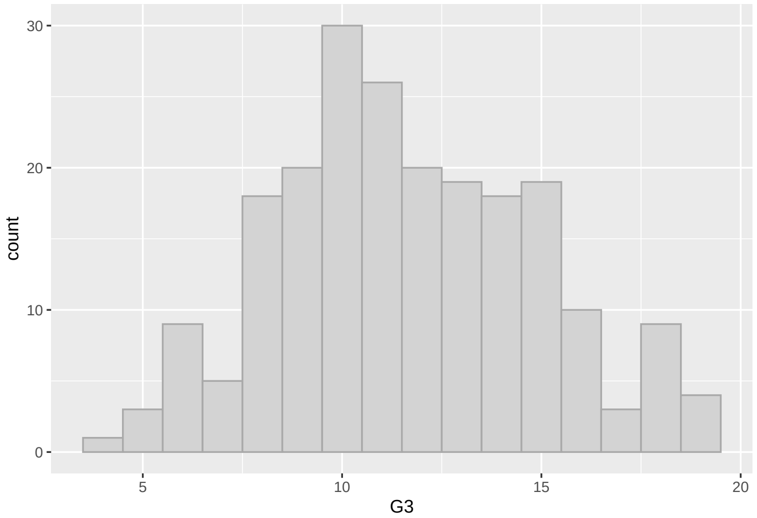
Task 3

Q: Plot the histogram of the final math grades in the training data. (0.5 points)

```r
# Install if missing & load package: ggplot2
packages <- c(packages, "ggplot2")
invisible(sapply(packages, install_if_missing))
```

```
## Skipping already installed package: glmnet
## Skipping already installed package: corrplot
## Skipping already installed package: ggplot2
```

```r
library("ggplot2")

# Create plot
ggplot(train) +
  geom_histogram(aes(x=G3), binwidth = 1, color = "darkgrey", fill = "lightgrey")
```



Task 4

Q: Explain shortly the difference between causal and predictive modelling. (0.5 points)

A: In predictive modeling, we infer the values of the dependent variable based on the values of the independent variable for a specific or multiple observations. The goal is to develop a formula for making predictions about the dependent variable. In contrast, causal modeling is interested in the degree to which a certain variable actually causes another variable to change, not just move in a certain correlation to it. This means that predictive modeling is much more concerned with the estimation error, whereas the model error is the central focus of a causal model.

Task 5

Q: Choose five variables that you consider most relevant to predict the final grade. Estimate two models by OLS, the first with your chosen set of variables, the second including all first order interactions. Discuss the in-sample fit of the two models. (1 point)

```r
# Do a stepwise regression and add the variables with the highest sum of squares
base.mod <- lm(G3 ~ 1 , data = train)    # base intercept only model
all.mod <- lm(G3 ~ . , data = train)     # full model with all predictors
stepMod <- step(base.mod,
                scope = list(lower = base.mod, upper = all.mod),
                direction = "both", trace = 0, steps = 5) # step-wise algorithm
# note: trace = 1 shows step-wise procedure up to 5 steps

stepMod$call # Linear regression with 5 variables that have the highest sum of squares
```

```
## lm(formula = G3 ~ schoolsup + failures + absences + Walc + Medu,
##     data = train)
```

```r
# Model with 5 variables
model_5 <- lm(formula = G3 ~ schoolsup + failures + absences + Walc + Medu,
              data = train)

# model with all variables
model_all <- lm(formula = G3 ~ ., data=train)

# In-sample fit measure via MSE calulation
MSE_5 <- mean((train$G3 - model_5$fitted.values)^2)
```

```
MSE_all <- mean((train$G3 - model_all$fitted.values)^2)

print(paste("MSE 5 variables (In-sample):", round(MSE_5,4)))
```

```
## [1] "MSE 5 variables (In-sample): 7.7808"
```

```
print(paste("MSE all variables (In-sample):", round(MSE_all,4)))
```

```
## [1] "MSE all variables (In-sample): 7.0855"
```

A: The sample fit is assessed via the mean-squared error (MSE). According to theory, adding covariates will always reduce the in-sample MSE and improve the in-sample fit. Here we can see that the in-sample fit for the all-variables model is better than the five-variables model because the latter has a higher MSE.

Task 6

Q: Choose another five variables, add them to your variables set and generate two additional models analog to exercise 5). Split your data into a training and estimation sample. Plot both, the in-sample and out-of-sample fit. Which of the four model performs the best? (1 point)

```
install_if_missing("ggpubr")
```

```
## Skipping already installed package: ggpubr
```

```
library(ggpubr)

# Do a stepwise regression and add the variables with the highest sum of squares
base.mod <- lm(G3 ~ 1 , data = train)   # base intercept only model
all.mod <- lm(G3 ~ . , data = train)    # full model with all predictors
stepMod <- step(base.mod,
                scope = list(lower = base.mod, upper = all.mod),
                direction = "both", trace = 0, steps = 10) # step-wise algorithm
# note: trace = 1 shows step-wise procedure up to 10 steps

stepMod$call # Linear regression with 10 variables that have the highest sum of squares
```

```
## lm(formula = G3 ~ schoolsup + failures + absences + Walc + Medu +
##     famsup + studytime + sex + internet, data = train)
```

```
# Model with 5 variables
model_10 <- lm(formula = G3 ~ schoolsup + failures + absences + Walc + Medu
              + famsup + studytime + sex + internet,
              data = train)

# model with all variables
model_all <- lm(formula = G3 ~ ., data=train)

## In-sample fit measure via MSE calulation
MSE_10 <- mean((train$G3 - model_10$fitted.values)^2)
MSE_all <- mean((train$G3 - model_all$fitted.values)^2)

print(paste("MSE 5 variables (In-sample):", round(MSE_5,4)))
```

```
## [1] "MSE 5 variables (In-sample): 7.7808"
```

```
print(paste("MSE 10 variables (In-sample):", round(MSE_10,4)))
```

```
## [1] "MSE 10 variables (In-sample): 7.3277"
```

```
print(paste("MSE all variables (In-sample):", round(MSE_all,4)))
```

```
## [1] "MSE all variables (In-sample): 7.0855"
```

```
## Out-of-sample fit measure via MSE calculation
out_model_5 <- predict(model_5, newdata = test)
MSE_5_out <- mean((out_model_5 - test$G3)^2)
print(paste("MSE 10 variables (out-of-sample):", round(MSE_5_out,4)))
```

```
## [1] "MSE 10 variables (out-of-sample): 9.373"
```

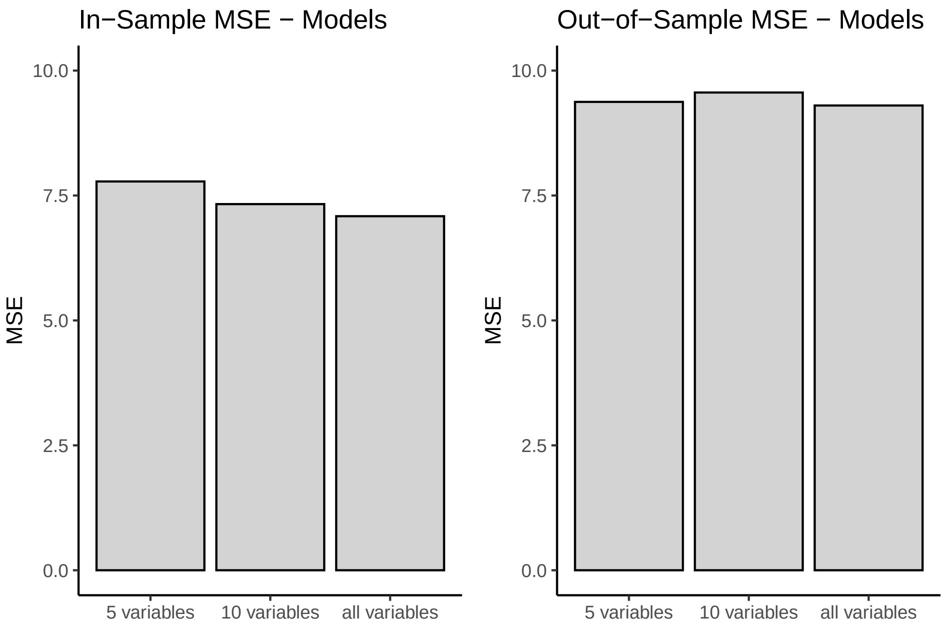
```
out_model_10 <- predict(model_10, newdata = test)
MSE_10_out <- mean((out_model_10 - test$G3)^2)
print(paste("MSE 10 variables (out-of-sample):", round(MSE_10_out,4)))
```

```
## [1] "MSE 10 variables (out-of-sample): 9.5606"
```

```
out_model_all <- predict(model_all, newdata = test)
MSE_all_out <- mean((out_model_all - test$G3)^2)
print(paste("MSE all variables (out-of-sample):", round(MSE_all_out,4)))
```

```
## [1] "MSE all variables (out-of-sample): 9.3009"
```

```
## Create dataframes for plots
MSE_in <- data.frame(model = c("5 variables", "10 variables", "all variables"),
                     MSE = c(MSE_5, MSE_10, MSE_all))

MSE_in$model <- factor(MSE_in$model,
                        levels = c("5 variables", "10 variables", "all variables"))

MSE_out <- data.frame(model = c("5 variables", "10 variables", "all variables"),
                      MSE = c(MSE_5_out, MSE_10_out, MSE_all_out))

MSE_out$model <- factor(MSE_out$model,
                       levels = c("5 variables", "10 variables", "all variables"))

# In-Sample Plot
in_sample_plot <- ggplot(MSE_in) +
  geom_col(aes(x=model,y=MSE),color="black", fill ="lightgrey") +
  ggtitle("In-Sample MSE - Models") + xlab("") + theme_classic() +
  ylim(0, 10)

# Out-of-Sample Plot
out_of_sample_plot <- ggplot(MSE_out) +
  geom_col(aes(x=model,y=MSE),color="black", fill ="lightgrey") +
  ggtitle("Out-of-Sample MSE - Models") + xlab("") + theme_classic() +
  ylim(0, 10)

ggarrange(in_sample_plot, out_of_sample_plot,
          ncol = 2, nrow = 1)
```



A: We can observe that the overall in-sample MSE is smaller for all models than the out-of-sample MSE. The in-sample MSE decreases, and the in-sample fit increases with an increasing number of variables. The out-of-sample MSE does not show a decreasing pattern. The ten-variables model has a higher MSE than the five-variables model, which is an indicator of overfitting. Hence, the ten-variables model has a worse out-of-sample fit than the five-variables model because overfitting constrains the model's usefulness outside of its original dataset. Adding too many covariates can lead to the model performing poorly for out-of-sample data because the high number of covariates in the model adjusts to the specified data set, thus resulting in the model being a worse predictor of other out-of-sample observations.
Adding all covariates to the model performs the best in- and out-of-sample.