Lab III: Regression. Burglaries and Facebook Interests

Achim Ahrens, Matthias Steiner

Version: 2023-03-03

We will use the Facebook data from Fatehkia et al. (2019). As a reminder: Their paper investigates the predictability of crime rates across US urban neighborhoods with the help of interests recorded on Facebook. They use the Facebook Advertising API to collect data on different interests of Facebook users of a particular ZIP code area. To this end, they combine the collected user data with demographic information and fit various statistical models to predict assault, burglary, and robbery rates as a function of the Facebook and census demographic variables.

Some details about the dataset you will be using:

- The list `Lab3_Variables.csv` contains a short description of each variable included in the dataset.
- Each observation represents a ZIP code area identified with the variable `zips`.
- The variables retrieved from Facebook are represented as ratios between the number of users with a specific interest and the total number of users older than 18 years in that ZIP code area. The variable `Drama_movies`, for instance, denotes the fraction of people interested in drama movies within the respective ZIP code area.
- The burglary rate corresponds to the number of burglaries per 100'000 residents. It is calculated by the number of burglary events reported in 2017 divided by the ACS 2015 population estimate multiplied by 100'000.
- Throughout this problem set, assume that the variable `Burglary_rate_2017` is the response or the target we want to predict and all the other variables represent features or covariates we use as inputs in our predictive models.

Laad the file `Lab3_data.csv` into R and make yourself familiar with the dataset. After you explored the dataset, please answer the following questions by providing the R code (if applicable) and by writing a short answer.

1 Preparation

1.1 One hot encoding

First, we create a dummy variable for each city. That is, we create a new indicator variable for each city name that is equal to 1 if the observation comes from the respective city and else 0. In the machine learning literature, the creation of dummy variables from categorical variables is often referred to as *one-hot encoding*. These newly created *dummies* can then be used in a statistical learning model.

This encoding step is required to make the data interpretable for the learning model. Note that the variable `City` is of *nominal scale*. Thus, neither the order of the categories nor their differences have any interpretable meaning. Therefore, it is not possible to keep `City` as one variable with different integer values. There are various functions available in R that produce dummies for you. However, for the sake of the exercise, build your own code for the encoding without using any other packages. Name the newly created covariates according to the city they represent (`Boston` for the city Boston, etc).

1.2 Missing values

Next, remove again all observation which contain missing values.

2 Linear regression: Part 1

2.1 Estimation

Consider the following regression model:

$$\texttt{Burglary\_rate\_2017} = \beta_0 + \beta_1 \texttt{Boston} + \beta_2 \texttt{Gospel\_music} + \beta_3 \texttt{Population\_median\_age\_2015} + \varepsilon \quad (1)$$

Estimate the model using ordinary least squares.

2.2 Mean Squared Error

The (in-sample) mean squared error (MSE) provides a measure of model performance. Calculate the MSE for your fitted model. Relate the MSE to the goodness-of-fit measure R^2. Do you think your results are trustworthy?

2.3 Prediction

Now assume that have a new observation with the following variable values:

- `Burglary_rate_2017`: 380
- `City`: Chicago
- `Gospel_music`: 0.1
- `Population_median_age_2015`: 40

What is the Burglary rate prediction of your model for this observation? Do the computation manually using the estimated coefficients *and* by using the `predict()` function.

2.4 Residuals

Next, you will create the residual plot of your fitted model. The residual plot is a scatter plot between the residuals and the fitted values of your model (put the residual on the y-axis of the plot). Add a horizontal line at the zero point of the y-axis. Add appropriate axis description and give your plot a title.

What is the interpretation of the points far away from the horizontal line you draw in the plot? How would it affect the MSE if you remove these points from your model?

3 Linear regression: Part 2

```
data3 <- data |> dplyr::select(Burglary_rate_2017,
                          Population_15_19_2015_perc:Heavy_metal_music)
```

You will now divide `data3` into a *training* (or estimation) sample and a *testing* sample (also called validation sample) with a 75:25 ratio. To be precise, 324 of the observations should belong to the training sample and the remaining 108 observations constitute the testing sample. You should assign the observations randomly in order to get representative samples. We use the *sample()* function for this:

```
set.seed(3)
train <- sample(c(rep(TRUE,324),rep(FALSE,432-324)), replace=F)
# training data
```

```
#data3[train,]
# testing data
#data3[!train,]
```

3.1 Model fits

Next, you will fit a series of linear regression models with `Burglary_rate_2017` as the dependent variable and *using the training data*. Start with a regression that includes a constant and the first predictor in your data set variable (i.e., `Population_15_19_2015_perc`). Then, add the second predictor (`Population_18_24_2015_perc`) and fit the regression model again. Repeat this procedure until all predictors from the dataset are included in the regression model.

There is an additional twist: instead of just including each predictor, we also want to include interactions of predictors. *Hint:* You can use the `(.)^2` function to create interactions in R formulas.

After fitting each model, compute the mean absolute errors (MAE) for the training and the testing data for each regression and store these values in two separate vectors.

3.2 Warning!

You might have noticed that a warning message pops up. Do you have any idea why the warning occurs? Look at the coefficient estimates of the models you fitted. What do you notice?

3.3 Training versus Testing Error

Use the previously retrieved MAE estimates and create a line plot with two lines for the testing and the training MAE. The y-axis should correspond to the MAE and the x-axis should be the number of features included in the model. Add a legend to your plot and give the axis appropriate descriptions. Also, add a title to your plot.

3.4 Bias and variance I/II

As you include more predictors, what do expect to happen to bias and variance?

3.5 Bias and variance II/II

Consider again the OLS estimates from Section 2.1. Suppose you define a new predictor that is formed by dividing each OLS coefficient by 2. How does bias and variance compare relative to the original OLS predictor? (No calculations required.)