Tutorial 8 - Linear Probability Model

AUTHOR

Victoria Hünewaldt

Tutorial 8

This tutorial will cover the linear probability model (LPM) and the root mean squared error. You will learn:

- how to run & interpret a LPM
- the problems coming with a LPM
- how to calculate & interpret the in-sample root mean squared error

Exercises

Create a .qmd-file and solve the tasks there. Store it in the JupyterHub folder “Session 8”.

Linear Probability Model

Create a model explaining “high_rating” (= binary endogenous variable) and run a linear probability model.

You first have to generate the dummy variable “high_rating” which should be equal to 1 if n_review_scores_rating>94 and 0 otherwise.

```
# clean environment & console
rm(list=ls())
cat("\014") # command to clear console

# load packages/libraries

# install.packages("remotes") # hashtagged because already installed
library(remotes)
remotes::install_gitlab("BAQ6370/sozoekds", host="gitlab.rrz.uni-hamburg.de")
library(sozoekds)

library(tidyverse)
library(dplyr)

airbnb_data <- airbnbsmall # store data as "airbnb_data"

#create a dummy variable for "high_rating"
airbnb_data$high_rating = ifelse(airbnb_data$n_review_scores_rating>94, 1, 0) # adds the varia
```

Choose the exogenous variables that you think explain “high_rating” most in your model. What can be problematic about linear probability models?

```
view(airbnb_data) # check again for variables

mod2 <-lm(high_rating~n_number_of_reviews+price+d_gym,data=airbnb_data)
summary(mod2)

# alternative: model including all variables as regressors
mod3 = lm(airbnb_data$high_rating~. ,data = airbnb_data) # the "." on the right side indicates
summary(mod3)

# R2 not applicable for binary response models

# problem of LPM: can predict negative probabilities or probabilities > 1
in_predictions <- predict(mod2, airbnb_data) # calculate predicted values
summary(in_predictions) # summary shows negative probability
```

Root mean squared error

Calculate the in-sample root mean squared error (RMSE). It measures the average difference between values predicted by a model and the observed values and provides an estimation of how well the model is able to predict the target value.

```
in_error <- (in_predictions-airbnb_data$high_rating) # calculate the error (predicted values -
sqrt(mean(in_error^2)) # calculate the RSME
# prints 0.494
summary(airbnb_data$high_rating) # mean: 0.478; ranges from 0 to 1

# alternatively, using function
RMSE <-  function(y,y_head){ # create a function that applies the RMSE formula
  RMSE = sqrt(mean((y-y_head)^2)) # RMSE formula
  return(RMSE) # specify RMSE as output of the function
}
rmse_mod2 = RMSE(airbnb_data$high_rating, in_predictions) # apply RMSE function
rmse_mod2 # prints 0.494
```