Data Analysis 2 - Assignment 2

Tunay Tokmak, Akos Almasi

2022-12-09

Let's clean the environment and import the required libraries.

```r
rm(list = ls())
library(tidyverse)
library(modelsummary)
library(marginaleffects)
```

Import the required data sets, filter for Barcelona, and merge them.

```r
hotels <- read_csv('hotels-europe_features.csv')
prices <- read_csv('hotels-europe_price.csv')
hotels_prices <- inner_join(hotels,prices, by = 'hotel_id')
hotels_barcelona <- hotels_prices %>% filter(city == 'Barcelona')
```

Normalize the price column. The price for each hotel must present the price per night for a correct analysis.

```r
hotels_barcelona <- hotels_barcelona %>% mutate(price_per_night = price / nnights)
```

Evaluate hotel ratings that have more than 5 commentor to have a balanced data.

```r
hotels_barcelona <- hotels_barcelona %>% filter(rating_reviewcount >= 5)
```

Evaluate hotels less than 5 miles far from center because there are many few observations more than 5 miles away from the center.

```r
hotels_barcelona <- hotels_barcelona %>% filter(distance <= 5)
```

Introduce the dummy variable

```r
hotels_barcelona <-  hotels_barcelona %>% mutate(highly_rated = ifelse(rating >= 4,1,0))
```

Split the data set into training and testing data sets. Our aim is prediction. Therefore, we split the data using 1/4 ratio.

```r
sample <- sample(c(TRUE, FALSE), nrow(hotels_barcelona), replace=TRUE, prob=c(0.8,0.2))
train  <- hotels_barcelona[sample, ]
test   <- hotels_barcelona[!sample, ]
```

Let's check the distribution of the dummy variable for categorical variables.

When we check the distribution based on accommodation type , we see that guest house and resort have many few observations. Therefore, we omit them.

```r
ggplot( )+
    geom_bar(data = train,aes(x = accommodation_type,fill = as.factor(highly_rated))) +
    scale_x_discrete(guide = guide_axis(n.dodge=2))+
    theme_bw()
```

```r
train <- train %>% filter(accommodation_type != 'Guest House',accommodation_type != 'Resort')
test <- test %>% filter(accommodation_type != 'Guest House',accommodation_type != 'Resort')
```

When the distribution based on weekend, holiday and month checked, the ratio of being highly rated does not change on the category. So we are not going to include them in the model.

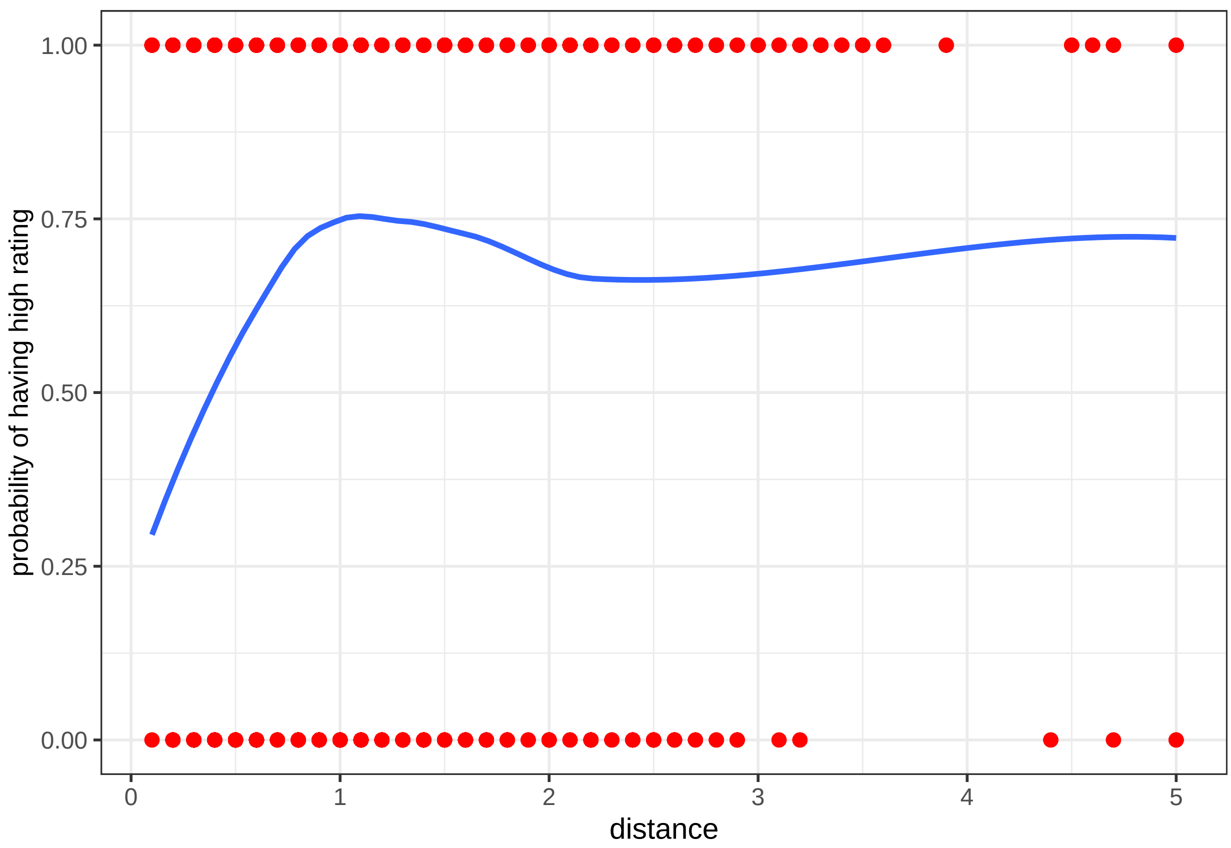
```r
ggplot( )+
  geom_bar(data = train %>% filter(highly_rated == 1),aes(x = weekend),
           fill = 'purple') +
  geom_bar(data = train %>% filter(highly_rated == 0),aes(x = weekend),
           fill = 'red', ) +
  theme_bw()

ggplot( )+
  geom_bar(data = train %>% filter(highly_rated == 1),aes(x = holiday),
           fill = 'purple') +
  geom_bar(data = train %>% filter(highly_rated == 0),aes(x = holiday),
           fill = 'red', ) +
  theme_bw()

ggplot()+
  geom_bar(data = train %>% filter(highly_rated == 1),aes(x = month),
           fill = 'purple') +
  geom_bar(data = train %>% filter(highly_rated == 0),aes(x = month),
           fill = 'red', ) +
  theme_bw()
```

Let's explore the correlation between the dummy variable and some numerical varibles Create the loess function.

```r
check_loess <- function(x_var, x_lab){
  ggplot( train , aes(x = x_var, y = highly_rated)) +
    geom_point(color='red',size=2,alpha=0.6) +
    geom_smooth(method="loess" , formula = y ~ x , se = FALSE)+
    labs(x = x_lab, y = "probability of having high rating") +
    theme_bw() +
    theme(axis.title.y = element_text(size = 10) )
}
```

The first graph demonstrates the correlation betweeen the distance and the dummy variable

```
check_loess(train$distance, 'distance')
```



The correlation does not look linear. Therefore we introduced a polynomial transformation, to capture the relationship better.

```
train <- train %>% mutate(distance_sqr = distance^2) %>% mutate(distance_cb = distance^3)
test <- test %>% mutate(distance_sqr = distance^2) %>% mutate(distance_cb = distance^3)
```

The second graph demonstrates the correlation between the distance_alter and the dummy variable.

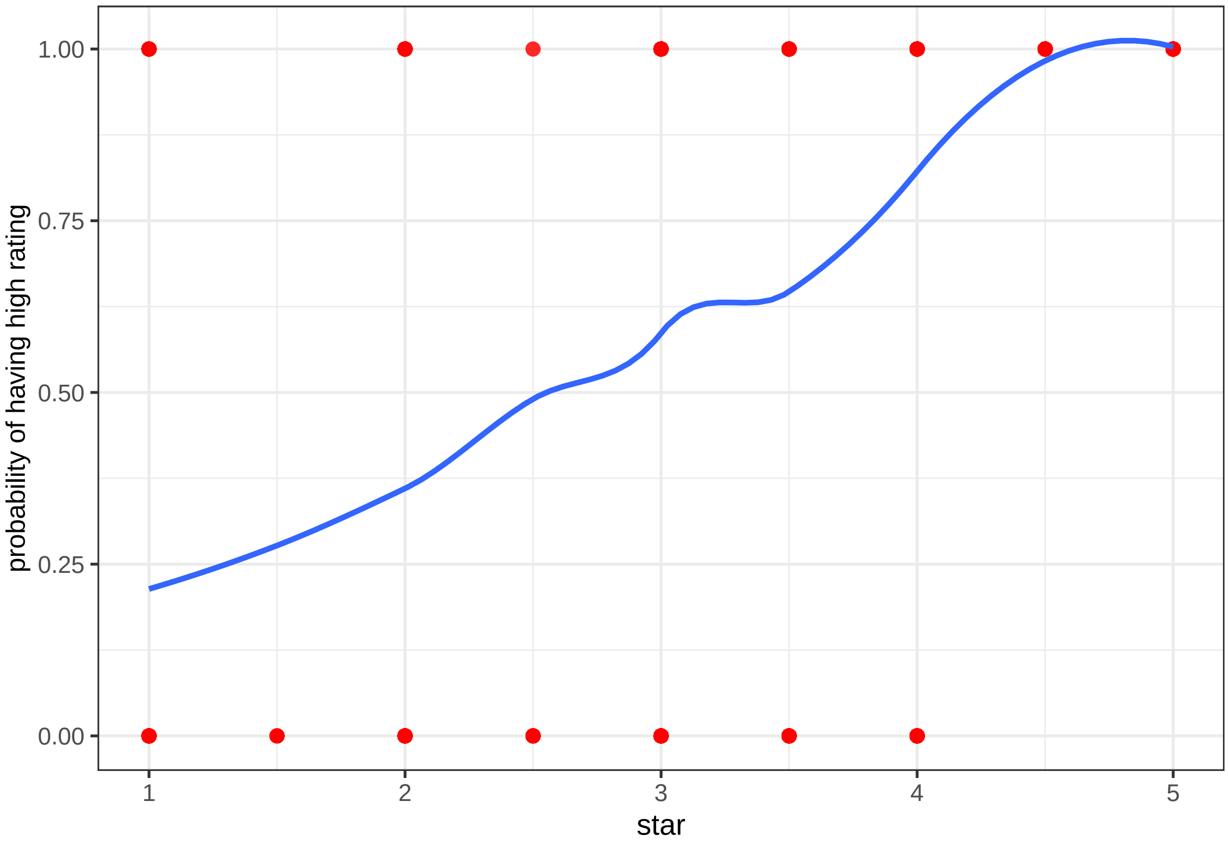
```
check_loess(train$distance_alter, 'distance_alter')
```

```
train <- train %>% mutate(distance_sqr_alter = distance_alter^2) %>%
  mutate(distance_cb_alter = distance_alter^3)
test <- test %>% mutate(distance_sqr_alter = distance^2) %>%
  mutate(distance_cb_alter = distance_alter^3)
```

The same principle applies to distance_alter variable. It affects the probability. However, we decided not to include it in our model. The reason is that the accuracy ratio for the test data set is higher without the distance_alter variable.

The third graph demonstrates the correlation between the stars and the dummy variable

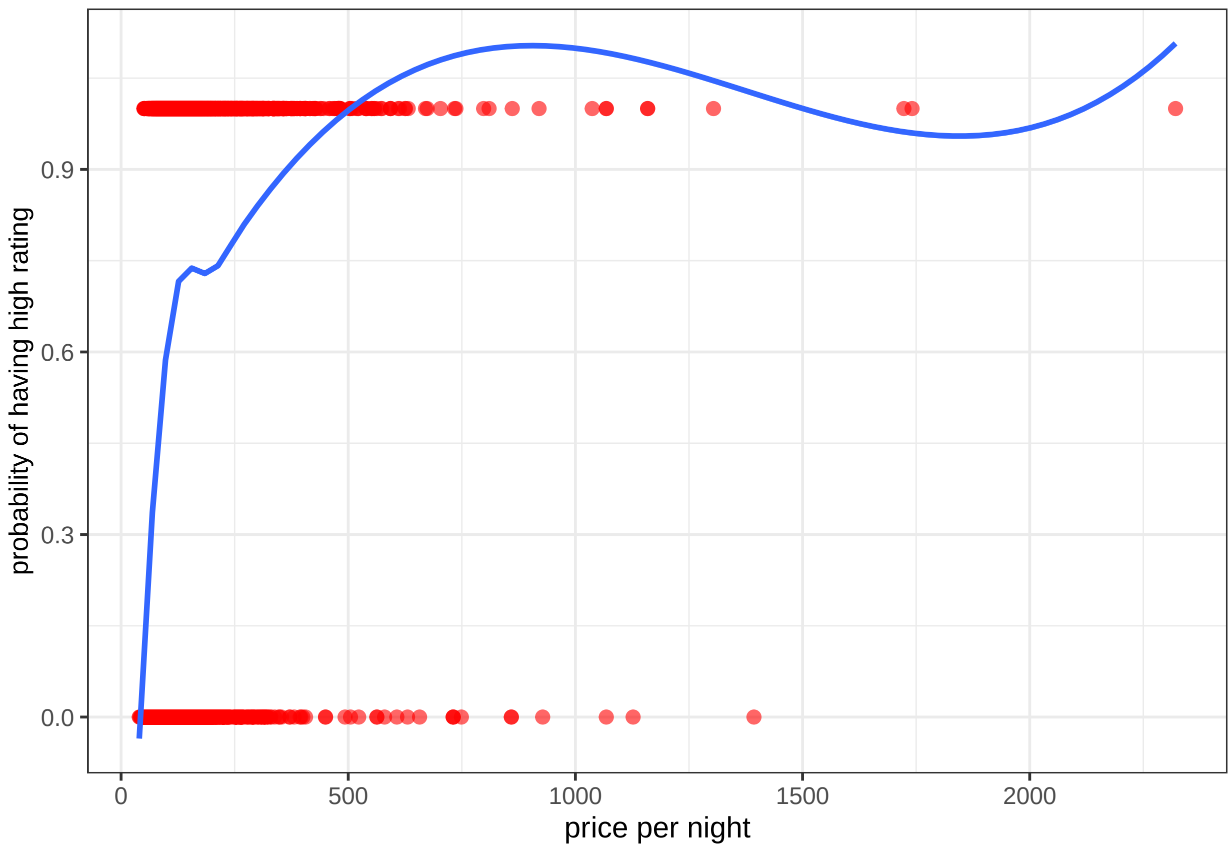
```
check_loess(train$stars, 'star')
```



The relationship is almost linear.

The fourth graph demonstrated the correlation between the price and the dummy variable

```
check_loess(train$price_per_night, 'price per night')
```



```
train <- train %>% filter(price_per_night < 2000)
test <- test %>% filter(price_per_night < 2000)
```

The graph looks like logarithmic which allows us to perform a natural logarithm transformation for the price_per_night variable.

```
train <- train %>% mutate(lnprice = log(price_per_night))
test <- test %>% mutate(lnprice = log(price_per_night))
```

```
lpm <-lm(highly_rated ~ distance + distance_sqr + distance_cb +
           #distance_alter  + distance_sqr_alter + distance_cb_alter +
           lnprice +
           stars ,
           data=train)

train$pred_lpm <- predict.glm(lpm, type="response")

logit <-glm(highly_rated ~ distance + distance_sqr + distance_cb +
              #distance_alter + distance_sqr_alter + distance_cb_alter +
              lnprice +
              stars  ,
              data=train, family = 'binomial')

logit_marg <- marginaleffects(logit)
```

```r
train$pred_logit <- predict.glm(logit, type="response")

probit <-glm(highly_rated ~ distance + distance_sqr + distance_cb +
               #distance_alter + distance_sqr_alter + distance_cb_alter +
               lnprice +
               stars ,
               data=train,
               family = binomial(link = 'probit'))

probit_marg <- marginaleffects(probit)

train$pred_probit<- predict.glm(probit, type="response")
```

```r
msummary(list( 'lpm' = lpm, 'logit' = logit, 'probit' = probit,'marg_logit' = logit_marg,
               'marg_probit' = probit_marg),
         gof_omit = 'DF|Deviance|Log.Lik.|F|R2 Adj.|AIC|BIC|R2|PseudoR2|RMSE',
         stars=TRUE,
         estimate = "{estimate}{stars} ",
         statistic = "std.error",
         output = 'markdown'

)
```

	lpm	logit	probit	marg_logit	marg_probit
(Intercept)	-0.727***	-7.018***	-4.154***		
	(0.072)	(0.456)	(0.262)		
distance	0.390***	2.080***	1.229***	0.357***	0.357***
	(0.055)	(0.311)	(0.184)	(0.052)	(0.053)
distance_sqr	-0.172***	-0.930***	-0.551***	-0.160***	-0.160***
	(0.029)	(0.168)	(0.099)	(0.028)	(0.028)
distance_cb	0.022***	0.119***	0.071***	0.020***	0.021***
	(0.004)	(0.024)	(0.014)	(0.004)	(0.004)
lnprice	0.113***	0.720***	0.416***	0.124***	0.121***
	(0.015)	(0.091)	(0.053)	(0.015)	(0.015)
stars	0.181***	0.929***	0.568***	0.159***	0.165***
	(0.008)	(0.049)	(0.029)	(0.007)	(0.007)
Num.Obs.	3675	3675	3675	3675	3675

When we check the models, we see that all intercepts and coefficients are statistically significant with a p value nearly 0. Distance is expressed as a polynomial. Therefore, one unit increase in distance will increase or decrease the probability of having a high rating based on the total of coefficients. However, we can say that the trend is as distance increases the probability decreases which is sensible. For price variable we see that the coefficients are around 0.11 for models. That means 10 percent increase in the price will increase the probability around 0.011 percent. Lastly stars coefficients are around 0.16 for the models. This leads to the conclusion that the stars variable is the most determining variable in the model. It makes sense that we already observed the sharp linear relationship previously. 1 unit increase in the stars will increase the probability of having high rating by 0.16 percent. The small effect of the price may be explained by the relationship of stars and price. However, we decided to keep it since it does not affect the prediction significantly.

```
test$pred_logit <- predict.glm(logit,newdata = test,  type="response")
test$pred_probit<- predict.glm(probit, newdata = test, type="response")
test$pred_lpm <- predict.glm(lpm,newdata = test,  type="response")

test$logit_prediction <- ifelse(test$pred_logit < 0.5, 0, 1)
test$probit_prediction <- ifelse(test$pred_probit < 0.5, 0, 1)
test$lpm_prediction <- ifelse(test$pred_lpm < 0.5, 0, 1)

mean(test$highly_rated == test$logit_prediction)
```

```
## [1] 0.7405765
```

```
mean(test$highly_rated == test$probit_prediction)
```

```
## [1] 0.7405765
```

```
mean(test$highly_rated == test$lpm_prediction)
```

```
## [1] 0.7372506
```

To evaluate the success of the models, we compare the accuracy scores based on predicted probabilities. All models perform ~72 percent correctly. Therefore, there is not a significant difference between the models.