Blocket Pricing

Evaluating Your Car’s Worth

Sebastian Strömberg

EC Utbildning

Kunskapskontroll - R

2024-04

Abstract

This study delves into regression modeling and statistical inference techniques to predict the value of used cars. Employing three distinct regression models, the logic method model, full regression model and the best subset regression model. The research aimed to identify the most influential variables and which model performed the best. The analysis concluded that the most significant variables were model age, mileage, and horsepower. Surprisingly the logic method model outperformed the other models, contrary to initial expectations.

Förkortningar och Begrepp

BIC – Bayesian Information Criterion

RMSE – Root Mean Square Error

Innehållsförteckning

1 Introduction

In the past year (March 2023 until February 2024), Sweden has seen an average of 25 070 newly registered cars every month, with a total of 300 844 registrations. There are almost 500 000 more cars in traffic than there were a decade ago, and this raises the question of what is happening with all the used cars as people keep buying new ones? Many of these cars get sold at a website called Blocket, but determining the market value of a used car remains a challenge. At this moment there are about 20 000 cars currently listed as for sale by private sellers on Blocket. Why not use this data and utilize a regression model to help determine a used car's worth?

This report aims to build a regression model capable of predicting the value of a used car. We will follow the steps from gathering and cleaning the data to developing a full model capable of making predictions.

The objectives and questions relevant to this are:

1. Build different regression models.
2. Evaluate which models perform the best.
3. Determine which variables are the most significant to predict a car's worth.

2 Theory

2.1 Model Evaluation

2.1.1 Root Mean Square Error (RMSE)

Root Mean Square Error is used to evaluate the accuracy of our prediction models. It measures the average deviation from the predicted values to the actual values.

$$RMSE = \sqrt{\sum_{i} (y_i - \hat{y}_i)^2}$$

2.1.2 Adjusted R Squared

Adjusted R-squared is a measure of the model's explanatory power adjusted for the numbers of predictors in the model. It is useful for assessing the overall fit of the model and it penalizes variables that do not fit the model. A higher adjusted r squared value indicates a better fit of the model to the data.

$$Adjusted\ R^2 = 1 - \left(\frac{n-1}{n-p-1}\right) \times (1 - R^2)$$

Where:

- n is the sample size
- p is number of predictors
- R^2 is the coefficient of determination

2.1.3 Bayesian Information Criterion (BIC)

Bayesian Information Criterion is a statistical measure used for model selection. It balances model fit with model complexity by penalizing models with more parameters. The lower BIC score signals a better model.

$$BIC = -2 \times \log(L) + k \times \log(n)$$

Where:

- L is the likelihood of the data given the model.
- k is the number of parameters in the model.
- n is the sample size.

2.2 Multiple Linear Regression

2.2.1 Theory Behind Regression Models

Regression modeling is used for looking at the relationship between one dependent variable and one or more independent variables. Here we use it to help predict the value (dependent variable) of a car based on multiple independent variables such as brand, mileage and age.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

Where:

- Y = dependent variable
- β_0 = y intercept (constant term)
- β_p = beta coefficient, the slope of the explanatory variable
- X_p = explanatory (independent) variable
- ε = Residuals error term

2.2.2 Logic method

When using the logic method, the analyst decides which independent variables to include in your model. In this case, 4 different variables were included, model year, mileage, horsepower, and engine size.

2.2.3 Full Regression Model

In the full regression method, all variables are included in the analysis. In this study, it contained 73 unique variables, ranging from different brands and types of cars to model years.

2.2.4 Best Subset Selection

Best subset selection aims to identify the most optimal model available while avoiding overfitting, instead of prioritizing accuracy alone which would lead to including all variables. Our model would then have a very low training error, but this method seeks to minimize but training and test errors. The algorithm behind the selection process is illustrated in the picture below.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Figure 1: Algorithm for best subset selection

3 Method

3.1 Dataset

The dataset used in this study was compiled through a combination of manual collection and web scraping from the Blocket platform, ensuring a homogeneous selection process. It consisted of 10,084 unique observations, each with 14 variables, within a price range from 20 000 SEK to 500 000 SEK. The reasoning behind this price range was to mitigate the influence of potential outliers and ensure the relevance of the model for everyday use. These variables included ID, brand, model, fuel type, gear, mileage, model year, car type, drive type, color, engine size, date of registration, region, and price. Using Excel, the dataset was structured into a table format with dimensions of 10,084 rows by 14 columns. Initially, the ID column was dropped from the data set.

3.2 Explanatory Data Analysis and Data Processing

After extensive testing of the dataset, which included 10,084 unique observations and 14 variables, further refinement was needed. We dropped all rows with missing values and car brands with less than 30 unique observations. Furthermore, the decision was made to exclude variables that were not deemed statistically significant or were too linear dependent. Model, color, region and date of registration were excluded from the dataset, leaving us with 8 dependent variables and one independent variable (price), and 9249 observations to work with. The reasoning behind this was that color and region were not statistically significant, date of registration was too linear dependent on model year. Additionally, the model variable was excluded to help streamline the regression model and make it easier to draw conclusions.

Furthermore, categorical variables such as brand, fuel type, drive type, car type and gear were transformed into dummy variables using one-hot encoding. This transformation enabled us to represent these categorical variables in a binary format, enhancing their usability and interpretability in our model. After transforming our categorical variables, we were left with 9249 observations and 73 variables where now each unique brand, fuel type, drive type, car type and gear is represented as a separate binary variable.

The last step before moving on to building our models was to split our data in training, validation, and test set. Sixty percent of the data was allocated for training, while 20% was used for validation, and another 20% for testing our final model.

3.3 Model Building

When constructing a regression model, there are various factors to consider. These include determining the statistically significant variables, assessing the normality in the data, identifying, and addressing potential outliers, and examining patterns for heteroskedasticity.

At first, both our dependent variable and the residuals displayed non-normal distributions, along with indications of heteroskedasticity. To address this, we applied a logarithmic transformation to our dependent variable, resulting in a distribution that approaches normality and addressing heteroskedasticity concerns. This adjustment aims to increase the statistical significance of our results.

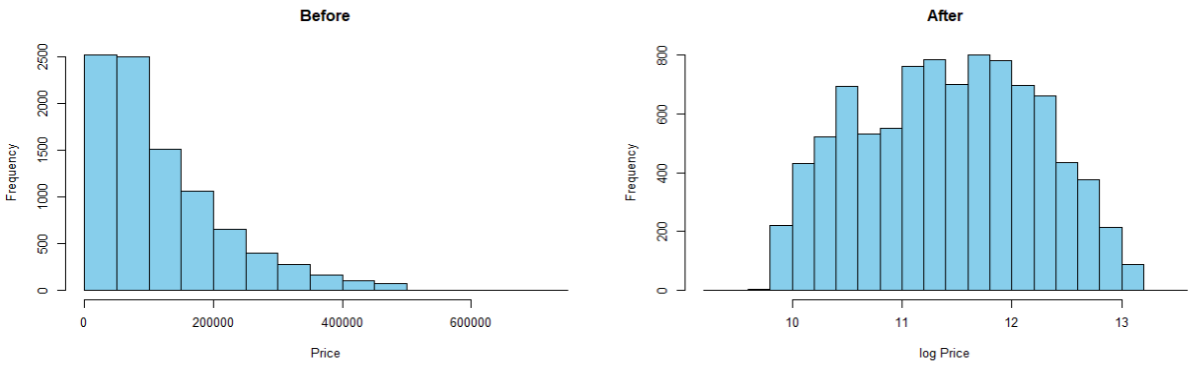

Figure 2: Before and after logarithmic transformations on Price

The next step was to examine potential outliers (observations that could drastically influence our models). Upon analyzing Cook's distance, we found that there was one observation that was highly influential and two others that were problematic. The decision was made to remove these three observations that all had a cook's distance exceeding 0.04. However, further testing revealed that removing these outliers made our models perform worse. Therefore, the decision was made to revert our decision and include these observations in our models.

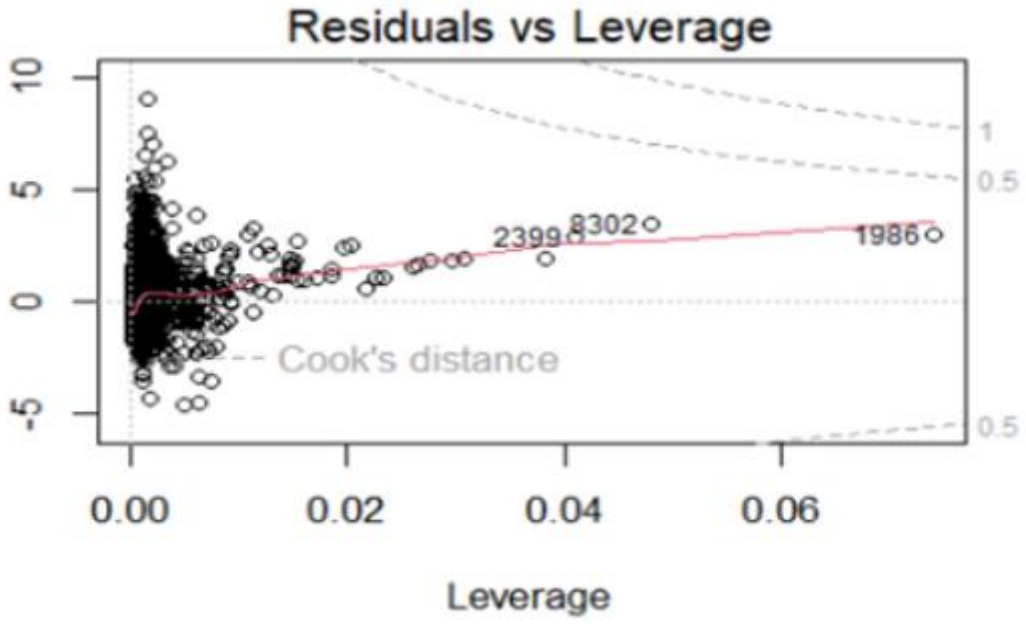

Figure 3: Potential outliers

3.3.1 Logic Method

For the logic method, the variables model year, mileage, horsepower, and engine size were selected. Upon evaluation the conclusion was determined that all variables chosen are statistically significant. With a p-value of less than <0.0000000002 we can discard the null hypothesis that our variables are not significant. The coefficients table below provides estimates for each variable, along with their standard error, t-value, and corresponding p-values.

Coefficients	Estimate	Std. Error	T Value	P Value	Significance
(Intercept)	-221.833060	2.278826805	-97.345	<0.0000000002	***
Model year	0.115498504	0.001130914	102.129	<0.0000000002	***
Mileage	-0.000003109	0.000000244	-12.744	<0.0000000002	***
Horsepower	0.004362648	0.000137340	31.765	<0.0000000002	***
Engine size	0.000115278	0.000013428	8.585	<0.0000000002	***

3.3.2 Full model

For our full model all 73 variables were included, leading to some issues. Warnings indicated that variables were too linearly dependent on each other, and some observations were too influential on our model. However, the purpose of our full model is to evaluate it on unseen data and compare its performance against our other models. Furthermore, as mentioned in the model-building step, removing the outliers worsened our model, so the decision was made to keep them. Below Is some of the most influential variables. As observed from our logical model, it appears as if we managed to select the most influential ones.

Coefficients	Estimate	Std. Error	T Value	P Value	Significance
(Intercept)	-214.9650571	2.3107620775	-93.028	<0.0000000002	***
Fuel Diesel	0.0462502790	0.0110726925	4.177	0.0000300	***
BrandPorsche	0.5344056113	0.1530884317	3.491	0.000485	***
Fuel Hybrid	-0.054780192	0.0129856565	-4.219	0.0000249883	***
Gear Manual	-0.151551440	0.0105329210	-14.388	<0.0000000002	***
Mileage	-0.000002655	0.0105329210	-12.491	<0.0000000002	***
Model year	0.0112180812	0.0011282622	99.428	<0.0000000002	***
Horsepower	0.0033079863	0.0001426387	23.191	<0.0000000002	***
Engine size	0.0000444789	0.0000133717	3.326	0.000886	***

3.3.3 Best Subset Selection

As mentioned in the theory part, the best subset selection aims to pick the most optimal model. In this case it ended up choosing only 3 variables, which were mileage, model year and horsepower. These variables show significant coefficients, as indicated by their low p value, showing a strong influence on the outcome. Below we can see the significance of each variable.

Coefficients	Estimate	Std. Error	T Value	P Value	Significance
(Intercept)	-212.7283999	2.0301759973	-104.78	<0.0000000002	***
Mileage	-0.000003066	0.0000002455	-12.49	<0.0000000002	***
Model year	0.1110046351	0.0010090208	110.01	<0.0000000002	***
Horsepower	0.0053531827	0.0000749807	71.39	<0.0000000002	***

Best subset selection keeps in mind that more variables are not always better, and in the end, we end up with a parsimonious model. The figure below demonstrates that after three variables the increase in adjusted r squared becomes insignificant, suggesting that using fewer variables prevents overfitting and results in a better predictive model.

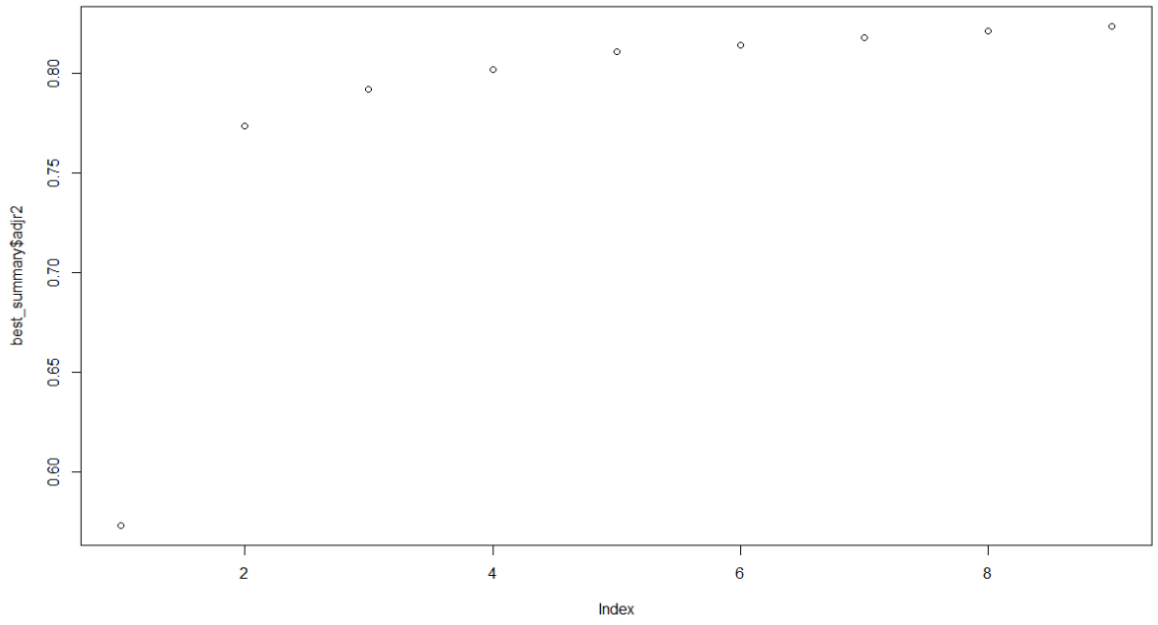

Figure 4: Y-axis show adjusted r-squared and X-axis how many variables

3.3.4 Summary models

It's interesting to see the performance of each model and how the variables influence the outcome. After comparing the results from the training-data I would not say we have a clear winner even though it looks like it is our full model if we're only looking at adjusted r squared and BIC. As mentioned earlier, when looking at adjusted r squared, we want a higher number and with BIC we want a lower number.

Our logic method has placed itself in the middle of our three models, both in adjusted r squared and BIC score but makes the best predictions. Our full model has the best adjusted R squared and BIC score, but it performs the worst when it comes to predictions. This is a sign that our full model is overfitted. Surprisingly the best subset selection model performs the worst in both adjusted r squared and BIC score but is in the middle when it comes to RMSE. Below are the results from the model building.

Results	RMSE	Adjusted R squared	BIC
Logic Method	56 690.72	0.783	4796.836
Full Model	65 050.54	0.911	3338.328
Best Subset Selection	61 044.27	0.779	4861.494

4 Results and Discussion

The logic model was chosen as our final model to predict on the test data. Its performance improved significantly, with a RMSE of 43 974.79, which is nearly 13 000 lower than on the validation data.

After conducting confidence and prediction intervals, we confirmed our model's consistency and practicality. In summary, the logic model outperformed the other models, making it this time the most suitable option for everyday predictions. See table below for intervals.

	Lower	Middle	Upper
Confidence interval	109 400.1	113 469.8	117539.5
Prediction interval	127 908.9	134 400.4	140 891.9

The conclusion was also drawn that our model is performing well between the interval 20 000SEK to around 250 000SEK. But it exhibits a notable deviation after this range, consistently overestimating prices, which also explains our high RMSE of 43974.79. Further investigation into the underlying factors could provide a valuable insight for model refinement. Addressing this offers an exciting opportunity for future research and potentially enhances our model's predictive accuracy.

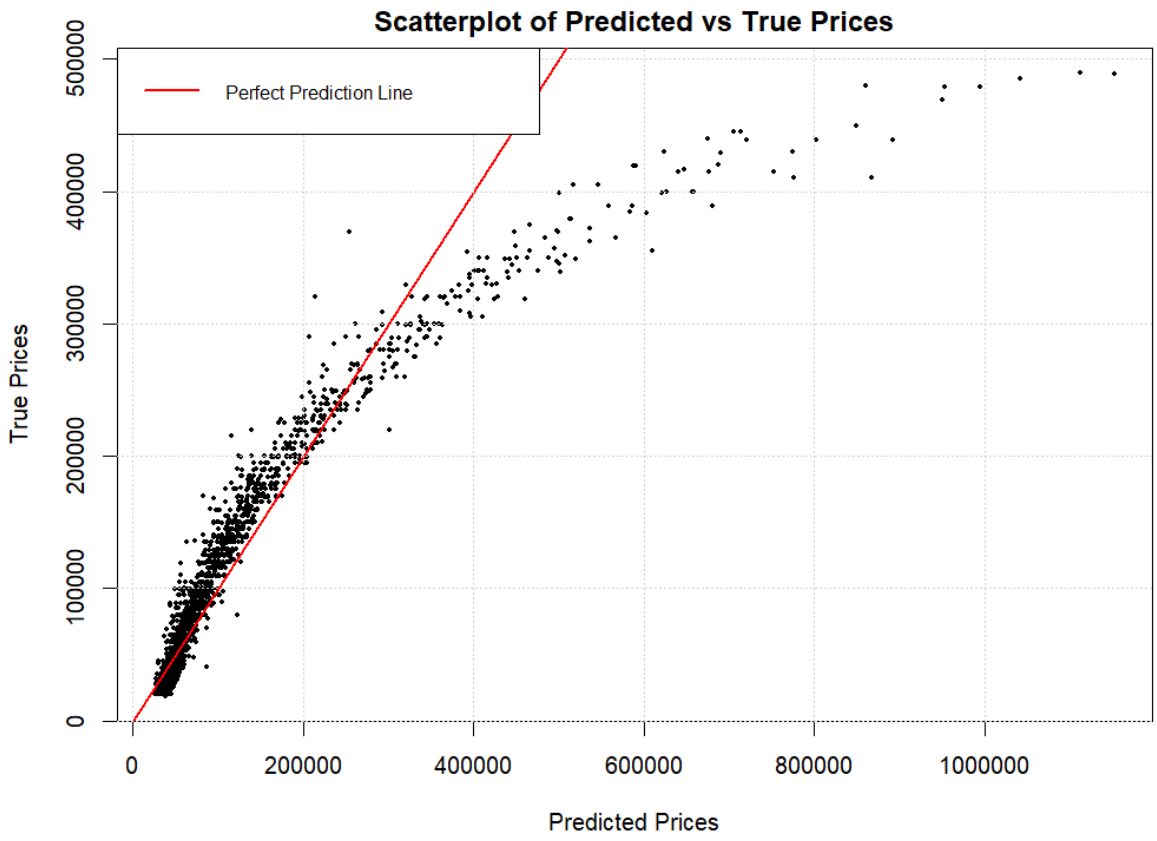

Figure 5: True prices vs predicted prices.

5 Conclusion

This study navigated the complexities of predicted used car prices through model construction and evaluation. Among the models explored, the logic model emerged as the standout performer, demonstrating better predictive accuracy. This was a surprise since this was the first model built and I, the analyst, decided which variables to use before investigating the variables significance. I thought that the best subset selection was going to emerge as the best model.

The significance of variables such as model year, mileage, and horsepower in determining a car's worth was evident. Providing us with valuable insight for model refinement. However, there was a notable deviation in performance at higher price ranges. In essence this research underscores the importance of all the steps needed to build a regression model in achieving accurate predictions.

For more detailed information on the data processing steps, model building techniques, and statistical analysis, please refer to the code provided in the appendix.

Appendix A

The R code for this report can be found below,

```
# Hämta API
# Ange URL till API:et
url <-
"https://api.scb.se/OV0104/v1/doris/sv/ssd/START/TK/TK1001/TK1001A/PersBilarDr
ivMedel"

# GET
res <- GET(url)
res

# utvinna informationen

rawToChar(res$content)

api_data = fromJSON(rawToChar(res$content))
names(data)


# Ladda in bildatan
file_path <- "C:/Users/sebbe/Desktop/Skolsaker/R/Datasets/bil_data_ren.csv"
bil_data <- read.csv(file_path, sep = ";", header = TRUE, fileEncoding =
"latin1", stringsAsFactors = FALSE)

head(bil_data)
dim(bil_data)
summary(bil_data)

# Ta bort alla NA rader
bil <- na.omit(bil_data)

# miltal är chr ändra den till numerisk (tack Tova för fina kodsnutten)

bil$Miltal <- gsub("[^0-9]", "", bil$Miltal)
bil$Miltal <- as.numeric(bil$Miltal)

head(bil)

# Ta bort bilmodeller med mindre än 30 observationer
bil_count <- bil %>%
  group_by(Märke) %>%
  summarise(count = n())

filtered_bil <- bil_count %>%
  filter(count >= 30)
```

```r
final_bil <- bil %>%
  filter(Märke %in% filtered_bil$Märke)

# log på pris för att få det mer normalfördelat
bil$Pris_log <- log(bil$Pris)


# One-hot encoding på våra kategoriska variabler
?dummyVars
encode <- c("Biltyp", "Drivning", "Bränsle", "Märke", "Växellåda")

#fullRank = TRUE för att inte hamna i dummy trap
encoded_df <- dummyVars("~.", data = bil[, encode], fullRank = TRUE) %>%
  predict(bil[, encode])

# Kombinera allt till ny df
df <- cbind(encoded_df, bil[, -which(names(bil) %in% encode)])
colnames(df) <- c(colnames(encoded_df), colnames(bil[, -which(names(bil) %in%
encode)]))

# lite EDA
head(df)
dim(df)
summary(df)

corr <- cor(df)
head(df)
print(corr)

# histogram före och efter log på pris

par(mfrow = c(2, 2))
hist(df$Pris, main = "Before", xlab = "Price", ylab = "Frequency", col =
"skyblue", border = "black",breaks = 25)
hist(df$Pris_log, main = "After", xlab = "log Price", ylab = "Frequency", col
= "skyblue", border = "black",breaks = 25)

# Splitta datan i train test val

spec = c(train = .6, test = .2, validate = .2)

set.seed(42)

g = sample(cut(
  seq(nrow(df)),
  nrow(df)*cumsum(c(0,spec)),
  labels = names(spec)
))
```

```r
res = split(df, g)

# Kolla så min split fungerade
sapply(res, nrow)/nrow(df)
addmargins(prop.table(table(g)))

# Nya variabler för train val test
# ta bort outlier
train_data <- res$train
val_data <- res$validate
test_data <- res$test


# Skapa 3 modeller och utvärdera och jämför. En där jag väljer variabler
själv, en Full och en Best

############## 1 Logic metoden,
# uppvisar mycket heteroskadicitet sak så gör LOG på pris
# logic <- lm(Pris ~ Modellår + Miltal + HK + Motorstorlek, data = train_data)
logic <- lm(Pris_log ~ Modellår + Miltal + HK + Motorstorlek, data =
train_data)
summary(logic)

par(mfrow = c(2, 2))
plot(logic)

vif(logic)

# Beräkna cooks avstånd då vi har några outliers jag vill ha bort över avstånd
0,04
#cooks_dist <- cooks.distance(logic)
#outlier_index <- which(cooks_dist > 0.04)

# Ta bort outliers och skapa ny datafram called train_datan
# train_datan <- train_data[-outlier_index, ]

#logic <- lm(Pris_log ~ Modellår + Miltal + HK + Motorstorlek, data =
train_datan)
# summary(logic)

# par(mfrow = c(2, 2))
# plot(logic)
# vif(logic)

############  2 Full
# Log så remove pris från df
# full <- lm(Pris ~ ., data = train_data)
full <- lm(Pris_log ~ .- Pris, data = train_data)
```

```r
summary(full)

alias(full)
vif(full)

par(mfrow = c(2, 2))
plot(full)
str(full)

#3 Best
############## 3 Best
?regsubsets

# best <- regsubsets(Pris ~ .,data = train_data, really.big = TRUE)
best <- regsubsets(Pris_log ~ .- Pris, data = train_data, really.big = TRUE)

summary(best)
best_summary = summary(best)
summary(best_summary)

names(best_summary)
best_summary$adjr2

par(mfrow = c(1, 1))
plot(best_summary$adjr2)

coef(best, 3)
plot(best, scale = "adjr2")

par(mfrow = c(2, 2))
plot(best)

# Skapa ny modell från best
best_predictors <-
names(which(best_summary$which[which.max(best_summary$adjr2), ] == 1))

# Lägg till "Pris" i listan över bästa prediktorer
best_predictors <- c("Pris_log", best_predictors)

# Välj kolumnerna som finns både i train_data och best_predictors
selected_columns <- intersect(names(train_data), best_predictors)

# Passa en linjär regression med endast de bästa prediktorerna
# best_model <- lm(Pris ~ ., data = train_data[, selected_columns])
best_model <- lm(Pris_log ~ ., data = train_data[, selected_columns])

summary(best_model)
```

```r
b_summary = summary(best_model)
par(mfrow = c(1, 1))
plot(b_summary$adjr2)

coef(best_model, 3)
plot(best_model, scale = "adjr2")

par(mfrow = c(2, 2))
plot(best_model)

# Beräkna VIF för den bästa modellen
vif_value <- car::vif(best_model)
print(vif_value)

# VIF resultat
# Miltal Modellår       HK
# 1.049186 1.053241 1.004528

# efter jag tagit bort outlier.. Sämre resultat efter jag tagit bort outliers,
gör om igen med outliers i min modell
#print(vif_value)
#Modellår Motorstorlek         Pris
#2.758531     1.594165     2.791296


#########  Utvärdera modellerna på validation datan och jämföra

val_pred_logic <- predict(logic, newdata = val_data)
val_pred_full <- predict(full, newdata = val_data)
val_pred_best_model <- predict(best_model, newdata = val_data)

results <- data.frame(
  Model = c("Logic", "Full", "Best"),
  RMSE_val_data = c(rmse(val_data$Pris_log, val_pred_logic),
                    rmse(val_data$Pris_log, val_pred_full),
                    rmse(val_data$Pris_log, val_pred_best_model)),
  Adj_R_squared = c(summary(logic)$adj.r.squared,
                    summary(full)$adj.r.squared,
                    summary(best_model)$adj.r.squared),
  BIC = c(BIC(logic), BIC(full), BIC(best_model))
)

results


# exponera vår log variabel så den blir normal igen
val_pred_logic_exp <- exp(val_pred_logic)
val_pred_full_exp <- exp(val_pred_full)
val_pred_best_model_exp <- exp(val_pred_best_model)
```

```r
# Skapa en ny data.frame för att hålla resultaten med riktiga priser
results_exp <- data.frame(
  Model = c("Logic", "Full", "Best"),
  RMSE_val_data = c(rmse(exp(val_data$Pris_log), val_pred_logic_exp),
                    rmse(exp(val_data$Pris_log), val_pred_full_exp),
                    rmse(exp(val_data$Pris_log), val_pred_best_model_exp)),
  Adj_R_squared = c(summary(logic)$adj.r.squared,
                    summary(full)$adj.r.squared,
                    summary(best_model)$adj.r.squared),
  BIC = c(BIC(logic), BIC(full), BIC(best_model))
)

results_exp


#########  Utvärdera vår valda modell (logic) på testdatan
final_model <- predict(logic, newdata = test_data)
rmse(exp(test_data$Pris_log), exp(final_model))

predicted_prices <- exp(predictions)
true_prices <- exp(test_data$Pris_log)

# Scatterplot för predicted vs true prices
plot(x = predicted_prices, y = true_prices,
     xlab = "Predicted Prices", ylab = "True Prices",
     main = "Scatterplot of Predicted vs True Prices",
     col = "black", pch = 20, cex = 0.7)  # Anpassa färger och symboler

# Lägg till linje för perfekt förutsägelse
abline(a = 0, b = 1, col = "red", lwd = 2)

# Lägg till rutnät
grid()

# Lägg till en legend
legend("topleft", legend = "Perfect Prediction Line", col = "red", lty = 1,
lwd = 2, cex = 0.8)
options(scipen = 999)


par(mfrow = c(2, 2))
plot(final_model)

summary(final_model)

confint(logic)

conf_int <- confint(logic)
```

```r
### Skapa ki och pi
confidence_intervals <- predict(logic, newdata = test_data, interval =
"confidence", level = 0.95)
prediction_intervals <- predict(logic, newdata = test_data, interval =
"prediction", level = 0.95)

confidence_intervals
prediction_intervals

ci_exp <- exp(confidence_intervals)
pi_exp <- exp(prediction_intervals)

ci_exp
pi_exp

##### Skapa medel ki och pi
# Beräkna medelvärdet av alla observationer för att få ett
konfidens/prediktionsintervall
mean_value_ci <- mean(ci_exp)
n_ci <- nrow(test_data)

# Beräkna standardfelet för ki
se_mean_ci <- sd(ci_exp) / sqrt(n_ci)

# Beräkna medelvärdet för pi
mean_value_pi <- mean(pi_exp)

# observationer för pi och medelfel
n_pi <- nrow(test_data)
se_mean_pi <- sd(pi_exp) / sqrt(n_pi)

# Beräkna z-värdet
z_value <- qnorm(0.975)

# Beräkna ki och pi för medelvärdet
ci_mean <- c(mean_value_ci - z_value * se_mean_ci, mean_value_ci,
mean_value_ci + z_value * se_mean_ci)
pi_mean <- c(mean_value_pi - z_value * se_mean_pi, mean_value_pi,
mean_value_pi + z_value * se_mean_pi)

ci_mean
pi_mean
```

Källförteckning

Antonio Prgomet, (n.d) Video Linjär Regression, hämtad 20 April 2024, från

https://www.youtube.com/watch?v=NcxMuCG6FS8&list=PLgzaMbMPEHEyLy3NJ8tZqHBzoVcZlowX4&index=3&ab_channel=EducationTopicsExplained

Bildata Blocket (n.d), hämtad 25 april 2024, från https://www.blocket.se/

James, G. Witten, D. Hastie, T. Tibshirani, R. (2023) An Introduction to Statistical Learning with Applications in R, Second Edition.

Rdocumentation, Regsubset (n.d) Hämtad 20 Aril 2024, från https://www.rdocumentation.org/packages/leaps/versions/3.1/topics/regsubsets

Statistikdatabasen, (n.d) Hämtad 20 April 2024, från https://www.statistikdatabasen.scb.se/