Analyzing Car Prices

Using Linear Regression



Robert Westman

EC Utbildning

Kunskapskontroll R-Programmering

202404

# Abstract

With the increasing availability of data, predictive analysis offers valuable insights into a multitude of domains, such as pricing predictions based on a set of available variables. This study focuses on predicting car prices using data from Blocket.se, an open online marketplace. The dataset includes 2700 observations, each containing various car attributes. This data was used to set up a linear regression model capable of both prediction and statistical inference. The top variables included horsepower, model year and mileage. Using a validation set, the model achieved a *Root Mean Squared Error* (RMSE) of 37,000kr from a mean of 270,000kr. Results when predicting brand new listings was quite accurate, with a majority deviating less than 10% from the actual price.

Contents

[1 Intro 1](#_Toc164848546)

[1.1 Questions 1](#_Toc164848547)

[2 Theory 3](#_Toc164848548)

[2.1 Data transformation 3](#_Toc164848549)

[2.2 Multiple Linear Regression 3](#_Toc164848550)

[2.2.1 Logical variables 3](#_Toc164848551)

[2.2.2 Best subset selection 3](#_Toc164848552)

[2.2.3 Looking into normality 4](#_Toc164848553)

[2.2.4 Residuals vs Fitted 4](#_Toc164848554)

[2.2.5 Q-Q Residuals 4](#_Toc164848555)

[2.2.6 Scale-Location 4](#_Toc164848556)

[2.2.7 Residuals vs Leverage 4](#_Toc164848557)

[2.2.8 Combating non-normality 4](#_Toc164848558)

[2.3 Shrinkage methods 5](#_Toc164848559)

[2.3.1 Ridge Regression 5](#_Toc164848560)

[2.3.2 The Lasso 5](#_Toc164848561)

[2.4 Measurements/estimates 5](#_Toc164848562)

[2.4.1 RSS 5](#_Toc164848563)

[2.4.2 Mallows Cp 5](#_Toc164848564)

[2.4.3 BIC 5](#_Toc164848565)

[2.4.4 Adjusted R-squared 5](#_Toc164848566)

[3 Method 6](#_Toc164848567)

[3.1 Data collection 6](#_Toc164848568)

[3.2 EDA/Feature transformation 6](#_Toc164848569)

[3.3 Comparing models 6](#_Toc164848570)

[4 Results and discussion 7](#_Toc164848571)

[4.1 EDA 7](#_Toc164848572)

[4.2 Best subset selection 8](#_Toc164848573)

[4.3 Train/Test split 8](#_Toc164848574)

[4.4 Sampling real observations 8](#_Toc164848575)

[5 Conclusions 9](#_Toc164848576)

[5.1 Questions 9](#_Toc164848577)

[5.2 End note 9](#_Toc164848578)

[References 10](#_Toc164848579)

# Intro

The amount of data available in the world today is staggering, with numbers estimating as high as 44 zettabytes in 2020, even further accelerated by the COVID-19 pandemic (Debanjan, 2020). This is, of course, because data have so many use cases. Global companies may collect data to get a grasp of their customer base, learn insights about how to value a product or simply to sell the collected data to the highest bidder.

Likewise, parties not looking for explicit financial gain may for example use data to gain insights about how the next election will go, what factors influence the general populations health and wellness or what risk-factors to look out for when trying to combat crime rates.

But one doesn’t have to be a global company to enjoy the benefits of the data available, anyone with knowledge about statistics and programming and a determination can use these techniques to make smarter decisions in everyday life. When browsing for a car to purchase, for example.

A graph of different colored dots

Description automatically generatedThis data was fetched using SCB’s (Statistiska Centralbyrån) API and shows newly produced cars running on gasoline, diesel and electric measured on a monthly basis from 2006-2024. A clear trend has emerged lately where EV’s have surpassed both the traditional fuel types. This raises questions about the valuation of EV’s compared to gasoline/diesel, something we will look into later.

Now for the purpose of this report, we will look at the data of cars for sale, analyze what influences the price and create a model that predicts the value of a previously unseen car. To achieve this, a few questions will be answered.

## Questions

1. How many observations are needed?
2. Can I automate the data collection process?
3. What factors drive the price of the car?
4. Are electric vehicles more expensive? Can I quantify by how much?
5. Does it make sense to include all cars, or do we need to filter our results?
6. What variables will we include in the model?
7. Can the model achieve good results? (+-10% from the real price)

Most of the theory practiced in this study stems from the book “An introduction to statical learning 2” written by James, G et al. (2022) and assumes basic knowledge about statistics, machine learning and programming. For reference, the code file used is available on my GitHub. If at any point I had a question to which I did not find a satisfactory answer, I consulted ChatGPT (OpenAI, 2024), then double checked the answer against other sources.

# Theory

## Data collection

Logic was first used to determine what data we wanted to include when gathering it, and we decided to use *brand, horsepower, mileage, model year, transmission, fuel type, car type, drivetrain*, and *color* as *x*-variables. We omitted using motor size which is obviously correlated to horsepower, and car model since our goal was to make a general model not specific to a brand. Additional features such as high-end sound systems or rear-view cameras are also not included since there was no reliable way to collect this data. *Price* was the target variable *y*.

## Data transformation

Seeing as the data must be presented in a way that is easily interpreted by R, we used the tools available within Excel to transform the data (Excel Documentation, 2024) to lower-case *(=LOWER*), replace unnecessary spaces and letters with nothing (*=SUBSITUTE*) and make sure columns containing numbers had the correct datatype.

After the data was loaded into R, most of our columns still didn’t have the correct datatype, so the transformations *as.numeric(bildata$pris)* and *as.factor(bildata$bränsle)* was run on all columns to turn columns consisting of letters into “factor” type and number-columns into “numeric” type. (R Documentation, 2024)

***Note:*** transforming the data in this way imposes limitations, a variable will contain *levels,* meaning you won’t be able to directly call a sublevel of the variable (i.e. for removal from the model). This was fine for my purposes, but if one requires that possibility, *df <- model.matrix(~ . + 0, data = df)* combined with *df <- as\_tibble(df)* can be used to set up dummy variables and then convert it to a data frame. This is included in the code file but not used.

## Multiple Linear Regression

A multiple linear regression model takes *several* variables as input as opposed to the standard linear regression model. This model can be used to determine if there is a mathematical relationship between a select set of variables and can even help us understand how they interact with each other. This equation takes the form of , where Y is the target value, are coefficients, X is our independent variables, p is our total variables (predictors) and is an error term attempting to catch the difference between estimates and true values. This equation is run to acquire a value for our predicted y-values \hat{y} with the goal to minimize the sum of squared residuals; RSS = . The goal of the equation is to reduce the sum of squared residuals between the real values and the estimated values predicted by a model. This is called the *least squares* criterion.

### Logical variables

Since logic was used to deduct which variables we included in the data collection process, theoretically all of them should hold significance, with color being a possible exception. *vif(model)* can thenprovide the Variance Inflation Factor for the model, giving us a quick way to check for *collinearity* within the variables.

### Best subset selection

Performing best subset selection to find the best variables/predictors involves fitting a linear regression model for each of all possible combinations of *p,* starting with *M0*, an empty model. Then models of all combinations of 1 predictor is fit, then all models are fit, all the way until where *k* is the total number of predictors. The best model (indicated by RSS, BIC, Cp or R-squared, see ***2.5*** for info) may then be used on a validation or test set for evaluation. While best subset selection can be very resource heavy to run since the number of models fit rise exponentially, it was fine for the purpose of this project.

### Looking into normality

A group of graphs with numbers and lines

Description automatically generated with medium confidenceUsing the *plot()* function on a model along with *par(mfrow=c(2,2)* yields four plots that can help us inspect the normality of the data. *Figure 1* contains the results of such a plot.

### Residuals vs Fitted

Here we have the predicted *y* values \hat{y} on the y-axis and the estimated fitted values on the x-axis. The closer the red line is to 0, the more linear the data.

### Q-Q Residuals

Figure 1

This plot shows the data plotted against a “true” normal distribution, where deviations from the straight line indicate some sort of outliers.

### Scale-Location

This is basically the same plot as the upper left, but these values are standardized. The goal is the same, to keep fitted values randomly spread along the red line.

### Residuals vs Leverage

This plot helps us find influential datapoints, if there is a point that is clearly separated from the rest or within the *cook’s distance* area, this can be indicative of an outlier. *Hatvalues()* and *which()* can be used to inspect eventual outliers or deviating clusters.

### Combating non-normality

This data seems fairly normal, but there are a multitude of methods to adopt if it isn’t. Removing outliers or performing transformations to scale the data are common methods to increase normality.

## Shrinkage methods

Instead of removing variables altogether, an alternate approach is to shrink the coefficients associated with a variable. This may reduce variance in the model estimations.

### Ridge Regression

Ridge regression is similar to the least squares method that linear regression uses, but the coefficients are estimated differently, according to this equation.

A math equations with numbers and symbols

Description automatically generated

Where the first term is the same as in least squares, but the second term introduces the λ (lambda), which puts a *shrinkage penalty* on the coefficients when they are close to zero. Lambda can be adjusted for your particular needs, where λ = 0 is the same as the least squares estimate. The higher the lambda value, the bigger the *shrinkage penalty*. This penalty is not imposed on the intercept.

### The Lasso

The lasso has a similar formula to Ridge, with a small adjustment to the second term. A math equations with numbers and symbols

Description automatically generated

has been replaced by . The difference this makes for the model is that the Lasso forces some of the coefficients to be exactly 0 if the lambda value is high enough, resulting in a *sparse* model containing only a subset of the original variables.



## Measurements/estimates

### RSS

See *2.2 Multiple Linear Regression.*

### Mallows Cp

Mallows Cp estimates the test *mean squared error* by using the following equation:A black and white text

Description automatically generated where is an estimate of the *variance* of the error term . This imposes a penalty on the training RSS to adjust for the eventual test error. This penalty increases as the number of predictors increases. Smaller values are desired.

### BIC

The *Bayesian Information Criterion* looks similar to Cp: A close up of a sign

Description automatically generated.

The second term is replaced by with a log term and since log *n* > 2 for any *n* > 7, BIC tends to penalize models with many predictors, resulting in BIC favoring models with fewer predictors. Smaller values are desired.

### Adjusted R-squared

The adjusted R-squared equation is as follows: . This is to adjust for the fact the RSS always increases as the number of predictors increases. This equation imposes noise variables on larger models and in theory, this leads to models with the best adjusted R-squared having only relevant variables and omit noise variables. Higher values are desired.

# Method

## Data collection

Our first objective was to get a hold of the data, and to get proof of concept without getting bogged down in technical difficulties. Our first approach was to manually input 50 observations per person in a joint Excel document. This worked fine, it was very easy to spot any irregularities in the data, but it was very time consuming, and we quickly looked at ways to automate this process. Also, 300 samples split into 6 different brands probably wouldn’t produce the best results, so we opted to get more data. The group also set up a filter when searching, to achieve consistency; model year between 2014-2024, price between 150,000-500,000.

A screenshot of a computer

Description automatically generatedWe looked at web scraping using Python, but again, it is quite technical and may take a while to get functional. We found webscraper.io, a third-party application that works as a browser extension. This tool had a simple graphical user interface where the logic was quite simple; in a chained list of events, perform specified actions and save the data specified. *Figure 2* shows the data in Excel, after its been transformed.

Figure 2

## EDA/Feature transformation

After loading the data into R, a quick exploratory data analysis was made, such as checking for NA values and transforming columns into proper datatypes. The *plot()* function was also used, revealing the contents in *Figure 1*. In an attempt to make the data more normal, a logarithmic transformation was used on the *price* column. A test to log-transform the *mileage* column was also conducted, but this yielded very small results and was not included in the final iteration.

## Comparing models

The first logical step was to fit a linear model to the data and look at results such as variable significance, the adjusted R-squared value and power of the coefficients. Best subset selection was used to determine the most influential variables for the model and the data was then split into a training/test set to evaluate performance.

A test was also conducted to see if imposing a penalty on the coefficients had a positive effect using Ridge/Lasso models.

# Results and discussion

|  |  |
| --- | --- |
| **RMSE for different models** | |
| Multiple Linear Regression | 37,000 |
| Lasso | 38,500 |
| Ridge | 38,000 |

Table 1: Root Mean Squared Error (RMSE) for the evaluated models. Mean: 260,000

## EDA

*A group of graphs with numbers

Description automatically generated*The initial data showed promise without any alterations at all, but after a few outliers were removed and the values of *price* were standardized, the normality of the data looked marginally better, as seen in *figure 3.* I took an interest in the clusters forming in the 4th plot and after investigating it seems all those observations are *cabs* or *coupés* respectively.

The full summary is shown in *figure 4,* revealing a low p-value, relatively high adjusted R-squared and significance on most variables. ***Note:*** *if wanted,**removal of “märkevolvo” due to low stat. sig. is not straightforward, see Note on page 3.*

*Figure 7* shows more interpretable coefficients, measured in Swedish krona. For example, every extra unit of mileage reduces car price by 4.98kr. Note that *car type* is listed in relation to *cab,* making all factorial variables harder to interpret.

Figure 3

A yellow and orange pixelated background with black text

Description automatically generatedThe *vif()* check for collinearity yielded no results above 5 or 10, so no action was taken.A screenshot of a computer

Description automatically generated However, a correlation heatmap was set up anyway, for inspection purposes. See *figure 5.* It affirms the highest correlation is between *price/model year* and *price/horsepower.*

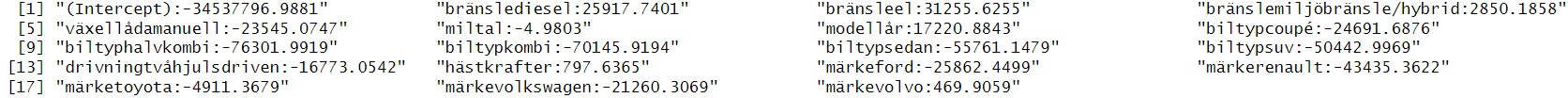


Figure 7

Figure 5

Figure 4

## Best subset selection

A group of graphs with numbers

Description automatically generatedThe results of the best subset selection mostly made sense, showing that *horsepower > model year > transmission* was the top 3 most influential variables. Remember that this is represented by the change in *price* per unit increase of *x* and all cars are affected by *transmission*. *Figure 6* shows the best number of variables given different criteria.

Figure 6

Seeing as all best results included 14-18 variables, it made sense to keep our current variables, I wouldn’t want to remove for example 1 car brand because it has low impact on the model, it still makes sense to include it.

If one wanted to extract the coefficients directly from the best subset selection, it’s possible, but dummy variables would have to be set up for this to work (*see* ***Note*** *on page 3*).

## Train/Test split

Evaluating the models required splitting the dataset into a train/test split, with a roughly 50/50 split between train and test data. The reason for the high train/test ratio was the relatively high number of observations available, training data of 1400 observations should be more than enough. The multiple linear regression model achieved an RMSE of 37,000 from a mean of 260,000. Not bad, but not as good as I had hoped.

The ridge/lasso models were tested in accordance with the book ISLR2 (p. 275~) and yielded RMSE’s of 38,000 and 38,500 respectively. A range of different *lambda* values was tested, between 5 and 1000, with similar results. This is not surprising; we already established our variables have low collinearity, so using these penalizing methods shouldn’t yield a better result, theoretically.

## Sampling real observations

|  |  |  |
| --- | --- | --- |
| Brand, model | Price (actual) | Price (predicted) |
| Audi A5 | 285,000kr | 291,000kr |
| Audi Q5 | 229,000kr | 212,000kr |
| Audi A4 | 169,000kr | 178,000kr |
| Audi Q3 | 279,000kr | 273,000kr |
| Audi RS3 | 369,000kr | 328,000kr |
| VW Golf | 200,000kr | 249,000kr |
| VW Golf | 239,000kr | 234,000kr |
| VW Tiguan | 239,000kr | 225,000kr |
| Audi E-Tron | 449,000kr | 637,000kr |
| Toyota Corolla | 245,000kr | 236,000kr |
| Toyota Corolla | 300,000kr | 272,000kr |
| Toyota RAV4 | 325,000kr | 367,000kr |
| Ford Kuga | 210,000kr | 230,000kr |
| Ford Mondeo | 205,000kr | 225,000kr |
| Ford Edge | 228,000kr | 246,000kr |
| Volvo V40 | 215,000kr | 221,000kr |
| Volvo XC60 | 278,000kr | 298,000kr |
| Volvo S90 T5 | 275,000kr | 240,000kr |
| Volvo XC60 T6 | 389,000kr | 475,000kr |

Ultimately the goal is to use the model to predict prices on new occurrences on blocket.se, so it makes sense to go there to test it out. See *table 2* for results.

Table 2

These new observations were picked randomly from the first page when using our blocket.se filtered search.

As seen, the linear model adapts quite well to new data, with many of the predictions falling within a 10% range of the actual price. Worth pointing out however is that it struggles with outliers such as high *horsepower* or if the car is very well equipped with features. Preferably, one could set up a filter that better specifies the range of interest to avoid this.

Important to note is that these prices are not “true” values, they are appraised and set by humans and could be under/overpriced.

# Conclusions

## Questions

1. Depends on the task. More data is generally better, but there is a point where it becomes redundant. 300 we felt was too few when split into 6 brands, so we extended to 2000+.
2. To an extent, yes. We kept it quite simple as to not get stuck on this step, the data collection was not fully automated but active time required to acquire the data was cut with roughly 90% by using the web scraper.
3. Almost all features we selected using logic were relevant for the price of the car. Our first iteration included the color as a feature, but as suspected this did not hold statistical significance and was cut moving forward.
4. Yes, they are. As seen in *figure 7,* EV’s are on average 31,200kr more expensive than gasoline cars. Combine this with the generally higher *horsepower*, four-wheel drive and high number of technological features, the price quickly ramps up.
5. For the model to make accurate predictions, filters need to be in place to avoid lumping old worn-out cars with new, fancy ones. This was apparent even in our filtered model, where brand new electric cars skewed the model because of their high *price* and *horsepower.*
6. In the end, *fuel type, transmission, mileage, model year, car type, drivetrain, horsepower* and *brand* were included.
7. It could not. I did not achieve RMSE within 10% when using a test data set, however when sampling real listings of cars, 12/20 fell within the desired 10% range. I think that if one defines the purpose and filters the training data better, this goal can be achieved.

## End note

While I think this linear model served its purpose quite well, I see several areas that can be improved. Central to this is the question; what is the purpose? If the purpose is to simply find an old daily driver, the data collection and modeling would have to exclude any cars not fulfilling these requirements. Likewise, if the purpose is to set prices for the dealership you work at, you would need to adjust the model for this, perhaps by segmenting your cars that are for sale and determining a price from the relevant features.

As for what could have been done better with this specific study, I think limiting the price further would’ve served the model well. Putting an upper limit of let’s say 300,000kr would have weeded out a lot of higher end cars and yielding better estimations for the mid-range cars. You may then create a separate model for high-end cars, if necessary.

# References

James, G., Witten, D., Hastie, T., Tibshirani, R. (2022). *An Introduction to Statistical Learning* (2nd ed.). Springer.

Debanjan, S. (2020). *How the world became data driven and what’s next.* Forbes.

<https://www.forbes.com/article>

Excel documentation (2024).

[https://support.microsoft.com/en-us/office/formulas-and-functions](https://support.microsoft.com/en-us/office/formulas-and-functions-294d9486-b332-48ed-b489-abe7d0f9eda9)

R documentation (2024).

<https://www.rdocumentation.org/>

OpenAI (2024). ChatGPT (GPT-3.5) [Large-Language-Model].

<https://chat.openai.com/>