Förutsäga Volvo Prislistor på Blocket med hjälp av regressionsanalys

Michael Barrett

Data Science – EC Utbildning

2025-04

Abstract

I det här projektet tränas en regressionsmodell på flera viktiga egenskaper hos Volvo-bilar som lades ut på den svenska webbplatsen Blocket. Den bästa modellen är en Lasso-regression som identifierade ungefär 92% av variationen i priset, isolerat inom fem nyckelförklarande variabler: ålder (kohorter), växellåda, miltal, drivning och hästkrafter. När modellen testades på osedda Blocket-data, överskattade den något bilpriserna, men var inom 3% av sin tränade prestanda. Detta tyder på att modellen visar en bra nivå av robusthet, men förutsägelser för både hög- och lågprissatta annonser inom datamängden visar att modellen lider av ett litet urval i dessa extrema fall och därför presterar under förväntningarna i dessa områden.

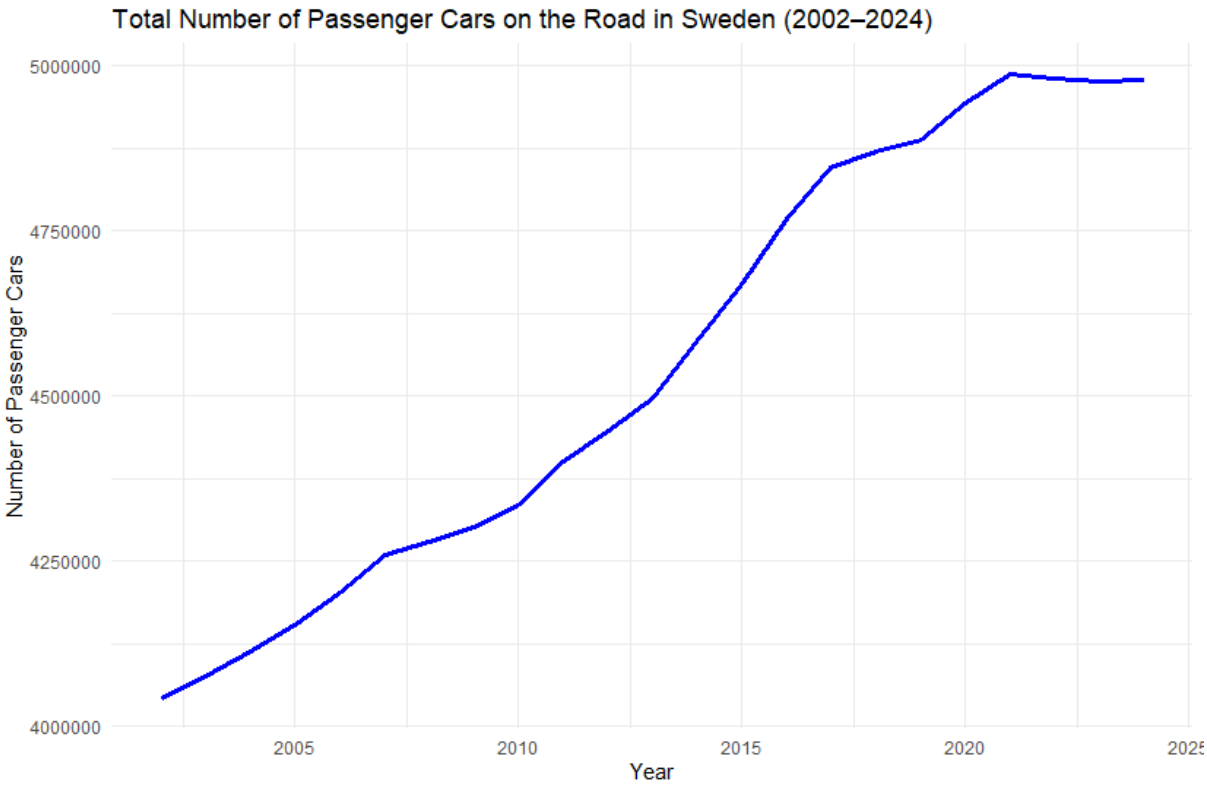
Skapas automatiskt i Word genom att gå till Referenser > Innehållsförteckning.

Innehållsförteckning

1 Inledning

Antalet personbilar på vägarna i Sverige har stadigt ökat och har ökat med cirka 25 % under perioden 2000 till 2025. Samtidigt har registreringarna av nya bilar i Sverige förblivit relativt konstanta under samma period. När nya bilar registreras säljs ofta gamla bilar vidare som begagnade via webbplatser som Blocket. Begagnatmarknaden för bilar är en stor och viktig marknad för de som inte har råd med eller inte vill köpa nya bilar på grund av de höga priserna.

Att bygga en regressionsmodell för att förutsäga pris kan vara fördelaktigt för både köpare och säljare. Köpare kan använda modellen för att identifiera om ett listat pris är 'bra' (om modellen förutspår ett högre pris än det listade priset) eller 'dåligt' (om modellen förutspår ett lägre pris än det listade priset). Säljare kan använda modellen för att föreslå listpriser och säkerställa att de inte underskattar värdet på ett fordon de planerar att sälja. Genom att ge dessa förutsägelser kan modellen hjälpa båda parter att fatta informerade beslut om sina köp.



Figur 1, Passenger Cars on the Road in Sweden between 2002-2024, (2025, SCB)

Målet med det här projektet var att identifiera Volvo-annonser på den svenska webbplatsen Blocket som är värderade avsevärt högre och avsevärt lägre än deras förväntade värde. För att uppnå detta mål identifierades flera delmål:

- **Identifiera** och **samla in** viktiga prisindikatorer för Volvo-bilar på webbplatsen Blocket.
- Träna en regressionsmodell med nyckelprisindikatorer från Blocket som kan fånga minst **90%** av variationen i bilpriser.
- Testa denna modell på osedda data för att utvärdera hur effektivt den presterar när det gäller att identifiera **exceptionella** (+/- %) Volvo-annonser på Blocket.

Jag satte även ett personligt mål att säkerställa att modellen inte använder mer än fem variabler.

Syftet med modellen är inte att ersätta mänsklig bedömning utan istället att effektivisera processen genom att flagga annonser som är exceptionella för vidare mänsklig bedömning. Detta är viktigt att påpeka, eftersom begagnade bilannonser kan variera kraftigt beroende på vissa faktorer som inte kan identifieras av traditionella maskininlärningsmodeller.

2 Teori

Introduktion

Blocket är den mest populära webbplatsen i Sverige för att köpa och sälja begagnade bilar. Bilannonser på webbplatsen ger information om bilens nyckelfunktioner och åtföljs också av ett listat pris. Webbplatsen har cirka 5 000 000 besökare per vecka, och 25% av bilannonserna leder till en försäljning inom 24 timmar (Blocket.se i Kadhammar & Wong (2021)). Detta gör Blocket till ett idealiskt val av webbplats för att samla in data.

Bilålder

När man bygger en modell som ska förutspå priset på en begagnad bil behöver det finnas en rättfärdigande för vilka och hur många variabler som inkluderas. En initial utforskande regression av flera variabler ledde till en tidig identifiering av 'Ålder' som en viktig prediktor för pris. I en modell där endast 'Ålder' förutspår 'Pris' uppnåddes ett R2 på 0,8. Detta innebär att denna modell identifierar cirka 80 % av variationen i 'Pris' genom den enda variabeln 'Ålder'. Detta innebär inte nödvändigtvis att den exakt identifierar 80 %, utan att det finns ett mycket starkt samband mellan 'Ålder' och 'Pris' i vår data. För att jämföra, en identisk modell som ersätter 'Ålder' med 'Bränsle' uppnår ett R2 på endast 0,23, vilket antyder att den identifierar cirka 23 % av variationen i 'Ålder'. Vid denna punkt blev det uppenbart att någon form av 'Ålder'-variabel måste inkluderas i modellen. Detta beslut förstärktes av akademisk konsensus som visade att ålder är en vital komponent för prissättning av begagnade bilar (Kumar & Sinha, 2024).

Miltal

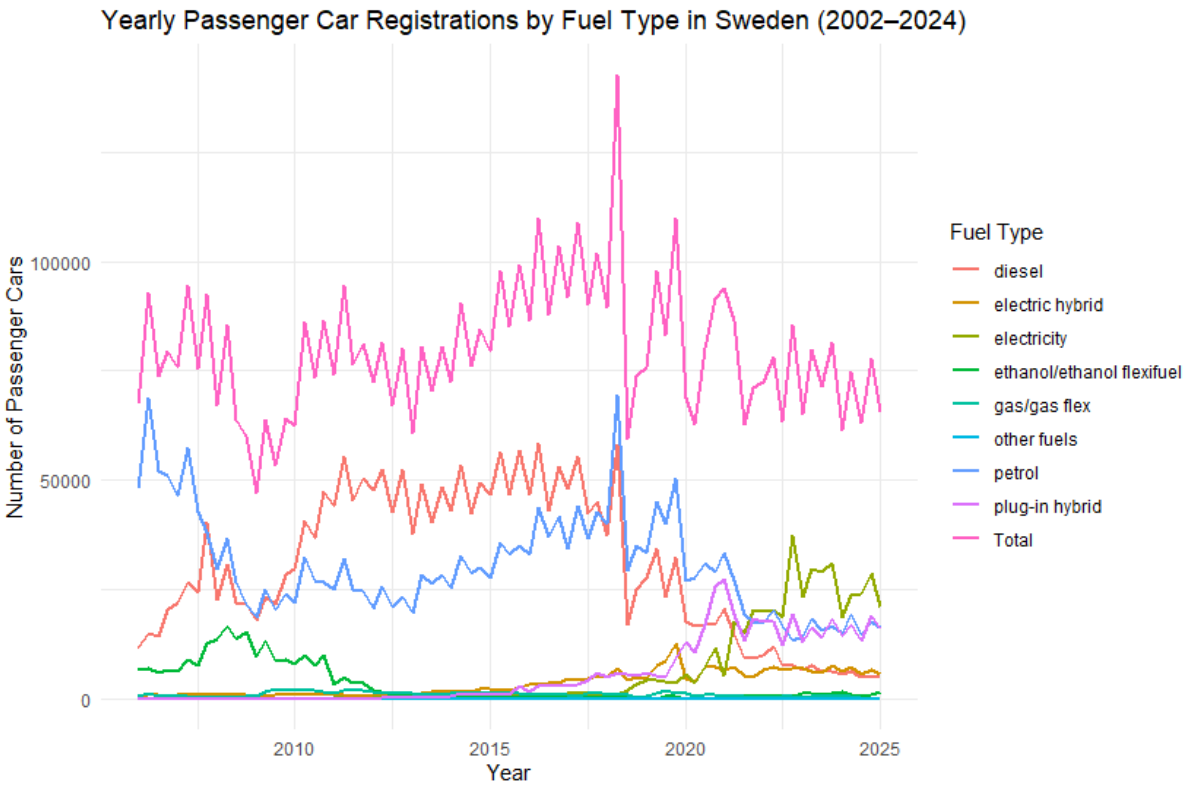
När man säger att bilens ålder förklarar 80% av variationen i pris, innebär det att modellen har förstått detta samband i ett vakuum. 'Miltal' anses ofta vara den näst viktigaste faktorn vid prissättning av begagnade bilar (Kumar & Sinha, 2024). Detta stöds när en utforskande regression görs med miltal som den enda prediktorn för pris, vilket ger ett R2 på 0,69~ (69% förklarande variabel). Så, hur kan då båda dessa variabler ge en så stark förklaring till listpriser? Här uppstår problemet med multikollinjäritet. Både ålder och miltal är korrelerade; när åldern ökar, ökar också miltalet. Detta innebär att det i en linjär regressionsmodell kan vara svårt att separera effekterna av varje individuell variabel på priset.

Den 'Bränsle' problem

Elektriska och hybridbilar har blivit oerhört populära de senaste 8 åren. Elbilar är nu den mest registrerade nya biltypen i Sverige varje år och har hållit denna position sedan cirka 2020 (SCB, 2025). Samtidigt har registreringarna av bensin- och dieselbilar fallit kraftigt – där bensinbilar fortfarande är ett populärt val för nya bilköpare (SCB, 2025). Dessa trender är viktiga när man står inför utmaningen att bygga en modell som använder bränsletyp. För el- och hybridbilar finns det ingen data där 'Ålder' sträcker sig långt över 5 år. Alla elbilar i datasetet uppfyller därför ofta ett relativt strikt kriterium. Elbilar har nästan utan undantag följande egenskaper: miltalet tenderar att vara lågt, åldern tenderar att vara låg, de är ofta automatiska och de har ofta en över genomsnittlig hästkraft.

När jag först byggde modellen, hade jag tänkt inkludera bränsletyp som en prediktor, eftersom jag logiskt antog att elbilar skulle listas till ett högre pris och diesel- och bensinbilar till ett lägre pris. En linjär regression med bara bränsletyp bekräftade detta. Men när ålder introducerades i modellen förändrades relationen mellan bränsletyp och pris avsevärt. Elbilar kopplades nu till lägre priser

jämfört med andra bränsletyper, när åldern hölls konstant. Denna omvändning drevs både av den dominerande effekten av ålder på pris och av bristen på äldre elbilar i datasetet. Denna brist på longitudinell data om elbilar innebar att modellen inte kunde särskilja effekten av bränsletyp från effekten av ålder. Detta är ett exempel på en förväxlingsvariabel och ledde till att bränsletyp uteslöts som en prediktor i modellen.



Figur 2, Yearly Vehicle registrations by Fuel Type 2002-2024, (2025, SCB)

Andra betydande prediktorer

'Växellåda', 'Drivning', och 'Hästkrafter' var tre viktiga egenskaper som ingick i modellen. Detta beslut togs baserat på användningen av en *best subset selection* i RStudio. Denna *best subset selection* kördes på alla egenskaper som bedömdes potentiellt användbara för modellen – 'age_cohort', 'Bränsle', 'log_Age', 'seller_dummy', 'Växellåda', 'sqrt_milage', 'Drivning', 'log_hästkrafter' (transformationer förklaras senare). *Best subset selection* identifierade en modell med 7 nyckelfunktioner, men ett respektabelt R2-värde på över 0,9 identifierades även med bara två funktioner. Det beslutades då att begränsa modellen till 5 nyckelfunktioner och ett R2 på 0,92, vilket förklarade 92% av variationen i pris. De tre funktionerna som lades till 'age_cohort' och 'sqrt_milage' var 'Växellåda', 'Drivning', och 'log_hästkrafter'. 'Bränsle' uteslöts från modellen efter att *best subset selection* identifierat det som en relativt svag förklarande funktion för priset samt de tidigare problemen som utforskades under rubriken 'The Bränsle Issue'.

3 Metod

Excel data städning

En dataset som innehöll 800 annonser från olika regioner i Sverige skapades av en grupp på 16 personer. Denna dataset innehöll 16 funktioner för bilar listade på Blocket under en period i mitten av april 2025. Dessa funktioner anges nedan:

Feature	Explanation
Säljare	Dummy: Privat / Företag
Bränsle	Categorical: Diesel / Bensin / Hybrid / El
Växellåda	Dummy: Automat / Manuell
Miltal	Numerisk: (t.ex 26 140)
Modellår	Numerisk: (t.ex 2013)
Biltyp	Categorical: Kombi / Halvkombi / SUV / Sedan
Drivning	Dummy: Tvåhjulsdriven / Fyrhjulsdriven
Hästkrafter	Numerisk: (t.ex 150)
Färg	Categorical: (...Grå / Blå / Röd etc.)
Motorstorlek	Numerisk: (t.ex 1984)
Datum_i_trafik	Date: (t.ex 2021-08-04)
Märke	Categorical: Volvo
Modell	Categorical: (...V40 / XC60 etc.)
Region	Categorical: (...Blekinge / Uppsala etc.)
URL (Referens)	Character: https://www.blocket.se/annons/###
Pris	Numerisk: t.ex 99 000

Figur 2, Table showing dataset columns - data available for modelling.

Denna data utforskades först visuellt i Excel, där formatet observerades och kolumnerna inspekterades kort för att se den allmänna kvaliteten på datan.

Kolumnerna ’Försäljningspris’, ’Miltal’ och ’Datum_i_trafik’ transformerades i Excel. ’Försäljningspris’ och ’Miltal’ transformerades för att förbättra det numeriska formatet, genom att ta bort mellanslag, så att ’99 999’ blir ’99999’. ’Datum_i_trafik’ transformerades för att skapa en ’Age’-variabel som skulle vara lättare att tolka och implementera i en modell. Denna variabel implementerades dock inte i den slutgiltiga modellen, utan en ’Age_cohort’-variabel användes istället, som grupperade bilar efter deras ’Modellår’ i anpassade kategorier.

Kolumner med N/A, saknad eller felaktig data städades också manuellt i Excel. Detta var endast möjligt på grund av datasetets relativt lilla storlek, och en mer grundlig städningsprocess i RStudio bör byggas om träningsdatasetet var större. Sammanfattningsvis var datakvaliteten hög, och därför drogs endast 14 observationer bort på grund av ofullständig data. Modellen byggdes därför på ett dataset bestående av 786 observationer.

Datatransformationer i RStudio

Många funktioner i denna analys var inte normalfördelade, så datan transformerades för att minimera påverkan av dessa fördelningar på modellens prestanda.

'Pris' visade sig vara höger-skev i sin råa form, och en log-transformation genomfördes för att motverka detta. Den resulterande 'log_Price'-funktionen blev vänster-skev, vilket fortfarande var problematiskt. Slutligen genomfördes en kvadratrots-transformation på 'Pris', vilket resulterade i en mycket mer normalfördelad 'sqrt_Price'.

Detta projekt har visat att bilens ålder är den viktigaste faktorn när man förutspår bilpriser på Blocket. Hur denna funktion implementeras i modellen är därför väldigt viktig. Ursprungligen var målet att inkludera 'Ålder' som en numerisk funktion, men detta ledde till problem med multikollinaritet med 'Miltal'. Det beslutades därför att ålder kunde transformeras till en kategorisk variabel, som fångar grupper av Volvo-bilar istället för rå ålder. Den nya 'age_cohort'-funktionen identifierar bilar utifrån perioden de tillverkades. Den använder 'Modellår'-funktionen för att placera observationerna i en av flera kategorier: '<2000', '2000-2009', '2010-2014', '2015-2019', och '2020+'. Valet av dessa kategorier baserades på fördelningen av bilar i dessa kohorter. Ett argument kan göras för att en ny kategori skulle kunna skapas som delar upp '2020+' i två – '2020-2022' och '2023+', då sambandet mellan ålder och listpris är mest tydligt för nya bilmodeller.

På samma sätt som för 'Pris' var 'Miltal' höger-skev i sin råa form. En identisk process som för 'Pris' genomfördes på denna funktion, vilket ledde till transformationen av 'Miltal' till 'sqrt_Miltal'. Denna transformation resulterade i en normalfördelad miltalsvariabel som var redo att implementeras i regressionsmodellen.

'Hästkrafter' var något höger-skev, men en log-transformation förbättrade denna fördelning, och därför blev 'log_Hästkrafter' den slutgiltiga funktionen som skulle inkluderas i modellen.

Slutligen inkluderades två dummyvariabler i modellen. **'Växellåda'** och **'Drivning'** visade sig ha en betydande effekt på priset under testningen av den bästa modellen. Dessa funktioner behövde inga komplexa transformationer, eftersom de redan var uppställda som dummyvariabler. 'Fyrhjulsdriven' och 'Tvåhjulsdriven' är funktionerna för 'Drivning', och 'Manuell' och 'Automatisk' är funktionerna för 'Växellåda'.

Modellval

Initialt byggdes en linjär regressionsmodell med de 5 funktionerna som nämnts tidigare. I 'age_cohort' användes '2020+' som referenskategori. Detta berodde på att denna kategori skulle associeras med de högsta prislistorna, och därför gör jämförelsen av de andra kohorterna mot denna kategori modellen lättare att tolka. Av samma anledning användes den funktion som var associerad med högre priser som referens för alla kategorivariabler. Detta innebär att 'Automatisk' användes i 'Växellåda', och 'Fyrhjulsdriven' användes i 'Drivning'.

En Lasso-modell kördes också för att jämföra prestanda. Båda modellerna uppnådde ett mycket liknande R2 (Linjär – 0.9209, Lasso – 0.9216) och RMSE (Linjär – 48.7419, Lasso – 48.7538), men AIC var avsevärt bättre i Lasso-modellen (Linjär – 8360.211, Lasso – 6128.022). Detta resultat innebar att den slutliga modellen skulle använda en Lasso-regressionsmodell, eftersom det avsevärt förbättrade AIC-värdet tyder på att Lasso-modellen borde prestera avsevärt bättre på ny data.

Träning och testdata

Modellen tränades på hela datasetet med 786 observationer. Detta innebar att ingen träning/testdelning genomfördes i datan. Ett nytt dataprovsurval togs från Volvo-listningar på Blocket som skulle användas för att testa modellen. Detta testprov bestod av 28 observationer och innehöll samma rådataformat som modellen tränades på.

4 Resultat och Diskussion

Lasso Regression Coefficients and Interpretations

Feature	Estimate (in Thousand SEK)	Interpretation
(Intercept)	222.35	Baseline price (2020+ model, automatic, 0 mileage, FWD, 0 horsepower
Ålder: 2015–2019	−42.92	2015–2019 models cost ~43k SEK less than 2020+
Ålder: 2010–2014	−123.03	2010–2014 models cost ~123k SEK less than 2020+
Ålder: 2000–2009	−211.74	2000–2009 models cost ~212k SEK less than 2020+
Ålder: <2000	−155.31	Older than 2000 cost ~155k SEK less than 2020+
Växellåda: Manuell	−29.76	Manual cars are ~30k SEK cheaper
Miltal (roten ur)	−1.63	Higher mileage reduces price (complex root to log comparison)
Drivning: Tvåhjulsdrift	−41.10	Two-wheel drive cars are ~41k SEK cheaper
Hästkrafter (log)	99.00	1% more horsepower increases price by ~990 SEK

Figure 1, Lasso Regression Coefficients with Interpretations.

Ovan är tabellen som visar koefficienterna för modellen. En tolkningskolumn har lagts till för att förbättra tydligheten, där den enda raden med begränsad tolkningsbarhet är raden för miltal. Detta beror på att interceptet (Pris) är formaterat i log-skala, och miltalet i en kvadratrot, vilket gör tolkningen svår utanför koefficientens riktning (för varje mil minskar priset).

En logistisk regressionsmodell med samma uppsättning funktioner genomfördes för att testa signifikans, där alla funktioner nådde en statistisk signifikansnivå på över 99,99%. I alla statistiska tester presterade Lasso-modellen bättre än den linjära modellen och valdes därför som den bästa modellen för projektet.

Model Testing	Linear Regression Model	Lasso Regression Model
R^2	0.92092	0.92160
RMSE	48.7419	48.7538
AIC	8360.211	6128.022

Testdata bestod av 28 nya Volvo-annonser på Blocket. Att köra modellen på dessa data ger oss en uppfattning om hur modellen presterar på osedda data. I genomsnitt underskattade modellen Volvo-annonserna i testdatan med -2,86%, men denna variationsnivå kan förväntas i ett relativt litet testset där annonser kan variera av ren slump. Sammanfattningsvis är detta ett bra mått på modellens prestanda, men fler metrik kan också härledas från detta.

MAPE (mean absolute squared error) för testsetet resulterade i 22,04%. Detta innebär att modellen i genomsnitt förutspår inom ungefär +/- 22% av vad säljaren faktiskt begär. Denna MAPE visar att det finns en ganska stor variation mellan vad vår modell förutspår och det verkliga listpriset, men detta bör ändå betraktas som en stark prestation. Detta beror på att vår modell förutspår det verkliga värdet på bilen baserat på dess nyckelfunktioner, men när en bil listas kan variation uppstå baserat på funktioner eller faktorer som ligger utanför modellens kontroll. Ett sådant exempel skulle vara hur snabbt någon vill sälja en bil. I slutändan definieras problemet av det faktum att värdet på en bil är subjektivt beroende av den individuella säljaren och köparen.

RMSE (root mean squared error) för testsetet resulterade i 33,21%. Detta innebär att när större misstag (bredare avvikelser mellan den förutspådda och verkliga prislistan) bestraffas hårdare, var modellens förutsägelser ungefär 33% felaktiga från det listade priset. Detta är intressant eftersom det belyser faktumet att det finns vissa förutsägelser som är mycket mer extrema, och att analysera var dessa förutsägelser misslyckas kan ge feedback för framtida förbättringar av modellen.

De största avvikelserna mellan det förutspådda och verkliga listpriset hittades i bilar med extremt höga priser (förutspått – 559 569 vs listat – 1 079 800 för en 48% avvikelse), och extremt låga priser (förutspått – 79 038 vs listat – 37 000 för en -114% avvikelse). Båda värdena är i de extrema ändarna av prisskalan, och det är där träningsdatan var bristfällig för modellen. Att ge modellen ett större dataset att träna på, samt att titta på icke-linjära regressionsmodeller skulle vara en bra riktning att gå i när man söker framtida förbättringar.

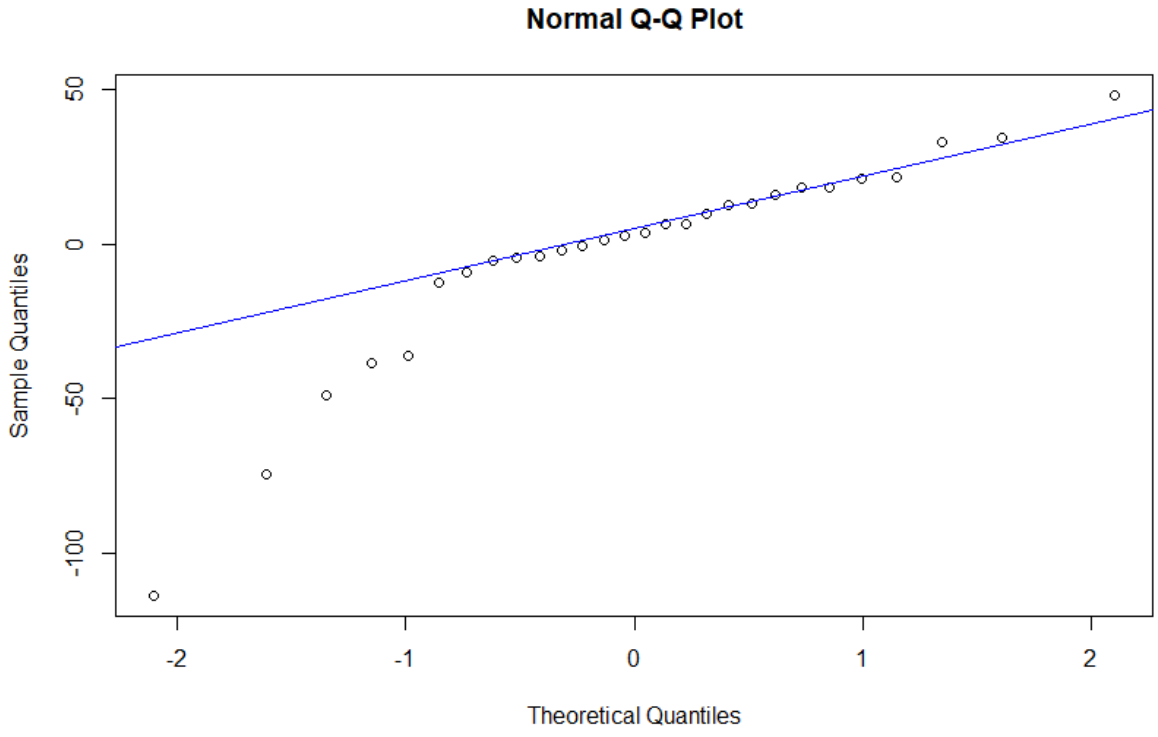


Figure 2, Q-Q Plot for Error Distributions in the Test Set.

Ovan finns Q-Q-plottet för felfördelningarna. Detta är en mer visualiserad form av det som diskuterades i föregående stycke. De flesta avvikelserna ligger på den negativa sidan (dvs. överskattad modell/underlistat pris).

Oavsett hur modellen kan förbättras, kan den information som denna modell ger ses som ovärderlig från både en köpares och säljares perspektiv. Om en köpare skulle kunna använda denna modell och inse att bilen som är listad för 37 000kr förutspåddes till 79 038kr, kan de titta närmare på denna annons och bedöma fordonets förtjänster för att se om det verkligen är ett "bra fynd". Samtidigt, om säljaren hade använt denna modell, skulle de ha kunnat värdera sin bil mer noggrant och lista den till ett högre pris.

Sammanfattningsvis känner jag att med denna modell har jag uppnått det jag satte upp för mig själv. Jag byggde en modell som förklarar över 90% av variationen i prislistor, och använde bara fem nyckelvariabler, och testade denna modell på verkliga data för att bekräfta dess effektivitet.

5 Slutsatser

1. En Quantile-Quantile (QQ) plot är ett sätt att visualisera fördelningar. På detta sätt är det mycket flexibelt, då det kan användas för att kontrollera fördelningar av beroende/oberoende variabler före analys, men också för att titta på hur residualfel är fördelade.

2. Inom regressionsanalys finns det en vikt vid inte bara förutsägelsen av resultat, utan också vid förståelsen av de underliggande relationerna mellan förutsägelserna. På ytan identifierar maskininlärning vissa funktioner som är kopplade till vissa resultat, men med statistisk inferens som följer med statistisk regressionsanalys är det också viktigt att förstå varför detta är fallet. Denna fråga om "varför" är avgörande, eftersom det möjliggör en djupare förståelse av relationen och om den är en legitim förutsägelse eller inte.

3. Den viktigaste skillnaden mellan ett konfidensintervall (CI) och ett förutsägelseintervall (PI) är nivån av säkerhet. Ett konfidensintervall fokuserar på medelvärdet – hur den genomsnittliga förutsägelsen faller inom modellens gränser. Ett förutsägelseintervall observerar istället hur en individuell observation kan falla inom modellens gränser.

4. I denna ekvation: '$Y = \beta 0 + \beta 1x1 + \beta 1x2 + \ldots + \beta pxp + \varepsilon$'.
Y i denna ekvation refererar till resultatet av modellen (i vårt fall i detta projekt, förutspått pris). B0 refererar till interceptet, vilket effektivt är värdet när alla andra funktioner i modellen hålls konstanta på '0', eller 'standard' kategorier. Varje b(1, 2, 3 etc.) refererar till en unik funktion inom modellen, där den respektive x(1, 2, 3) refererar till en enhetsökning i denna funktion och dess respektive inverkan på Y. I fallet med kategoriska eller dummy-variabler resulterar en enhetsökning i att den binära variabeln går från 0 till 1, så exempelvis 'växellåda' blir 'Manuell' (1) istället för 'Automatisk' (0).

5. Det är korrekt att det inte är nödvändigt att genomföra en tränings/testuppdelning av data vid användning av regressionsmodellering. Det kan dock vara både givande och hjälpsamt att testa en tränad modell på ny data för att se dess verkliga prestanda, som gjordes i detta projekt. Anledningen till att det inte är nödvändigt att genomföra en test/validerings/testuppdelning är att det finns statistiska metoder som är utformade för att hjälpa till att bestämma de bästa passande modellerna. Både BIC (Bayesian Information Criterion) och AIC (Akaike Information Criterion) användes i detta projekt för att hitta den bästa modellpassningen och funktionerna.

6. En bästa delmängdsurval genomförs för att bestämma den bästa kombinationen av funktioner att inkludera i en linjär regression. Det kan vara intensivt, särskilt när många frihetsgrader läggs till testet. Det är därför användbart att tidigt avgöra om några variabler ska uteslutas för att säkerställa att delmängdsurvalet endast väljer från funktioner som du är säker på att du vill inkludera i modellen. Den tar flera nyckelindikatorer för modellen som justerat R2, AIC och BIC och bestämmer den starkaste modellen. Ett exempel på ett problem som kan uppstå i delmängdsurval är att i kategoriska variabler med ett stort antal kategorier (som modell i Volvo – bestående av mer än 23 kategorier) kommer ett delmängdsurval att fånga specifika modeller som mycket relevanta för modellen, men att faktiskt implementera detta i modellen kan vara utmanande.

7. "Alla modeller är fel, några är användbara..." av George Box berör flera centrala problem med prediktiv modellering. För det första att våra modeller i slutändan är ett matematiskt mått på vad som är mest troligt att hända; alla modeller byggs med varierande grad av felvärden som håller den oberäknade variationen i resultat. Endast inom den renaste vetenskapens ämnen kan en modell ge en helt korrekt modell för förutsägelse, inom kemins byggstenar och inte mycket mer. Verkliga

världen är långt mer komplex än algoritmer och ekvationer. För det andra att vi som dataforskare bör sträva efter att bygga modeller som är robusta och kraftfulla, framför allt genom att förstå ämnet för förutsägelsen, rigorös testning och noggranna data.

6 Självutvärdering

1. Vad tycker du har varit roligast i kunskapskontrollen?

To see the model become a 'real' thing, with predictions that I ran on the test data. Seeing the coefficients and how they make sense. I also really enjoy the decision-making that comes with choosing features, and how it leads to transforming variables into certain forms to make the analysis stronger.

2. Hur har du hanterat utmaningar? Vilka lärdomar tar du med dig till framtida kurser?

I forgot how picky age can be as a variable. When I completed my Masters degree I remember it being 'difficult' to manage with collinearity and age, but when I came back to this assignment again, I was reminded once more that dealing with age in regression can be a fun challenge.

3. Vilket betyg anser du att du ska ha och varför?

VG hopefully, but again, as usual, I was a bit rushed on this assignment.

4. Något du vill lyfta till Antonio?

Nope. Thanks for the course. Was a nice one.

Appendix A

RStudio Code

(Some of the last lines of code may not run if ran now as I did a bit of cut/dropping to pull some visualizations)

```
library(readxl)
library(tidyverse)
library(scales)
library(dplyr)
library(stringr)
library(ggplot2)
library(broom)
library(car)
library(leaps)
library(glmnet)
library(gt)
library(knitr)

options(scipen = 999)

file_path <- "C:/Users/micha/Desktop/EC Utbildning som Data Scientist/07_R/Kunskapskontroll 1/Blocket_cars.xlsx"

excel_sheets(file_path)

blocket_data <- read_excel(file_path, sheet = "Sheet1")

## View(blocket_data)

## Summarizing the Age variable in the data.
```

```
blocket_data %>%
  summarise(
    n = n(),
    missing = sum(is.na(Age)),
    min_age = min(Age, na.rm = TRUE),
    q1 = quantile(Age, 0.25, na.rm = TRUE),
    median_age = median(Age, na.rm = TRUE),
    mean_age = mean(Age, na.rm = TRUE),
    q3 = quantile(Age, 0.75, na.rm = TRUE),
    max_age = max(Age, na.rm = TRUE),
    sd_age = sd(Age, na.rm = TRUE)
  )

## Histogram for Age of cars

blocket_data %>%
  ggplot(aes(x = Age)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "white") +
  labs(
    title = "Histogram of Car Age",
    x = "Age (years)",
    y = "Count"
  ) +
  theme_minimal()

## We can see that most cars are newer - makes sense.

## Plotting a small age vs price regression to check the general relationship.

ggplot(blocket_data, aes(x = Age, y = Price)) +
```

```
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm") +
  scale_y_continuous(labels = scales::comma) +  # Format y-axis to whole numbers
  theme_minimal() +
  labs(title = "Price vs. Car Age", x = "Car Age", y = "Price")

## As predicted, as age increases, price decreases. Obviously basic linear regression
## is predicting negative values, as it is linear.

## Transforming the data for seller into a dummy variable - 0 = Private, 1 = Company.

blocket_data <- blocket_data %>%
  mutate(
    seller_dummy = if_else(Säljare == "Företag", 1, 0)
  )

## Looking at the seller dummy.

blocket_data %>%
  summarise(
    mean_seller = mean(seller_dummy),
    n_foretags = sum(seller_dummy),
    n_privat = sum(seller_dummy == 0),
    total = n()
  )

## So we can see that the mean is 0.773, meaning 73.3% of listings are company.
## No missing data.

## Checking mileage now.
```

```
blocket_data %>%
  summarise(
    n = n(),
    missing = sum(is.na(Milage)),
    min_milage = min(Milage, na.rm = TRUE),
    q1 = quantile(Milage, 0.25, na.rm = TRUE),
    median_milage = median(Milage, na.rm = TRUE),
    mean_milage = mean(Milage, na.rm = TRUE),
    q3 = quantile(Milage, 0.75, na.rm = TRUE),
    max_milage = max(Milage, na.rm = TRUE),
    sd_milage = sd(Milage, na.rm = TRUE)
  )

## Done - so now we have Seller, milage, and age that are clean.

## The other I want to include is fuel - which may need to be coded
## as one-hot.

blocket_data %>%
  count(Bränsle, sort = TRUE) %>%
  mutate(
    proportion = n / sum(n)
  )

## Lots of untidy data - cleanup.

blocket_data <- blocket_data %>%
  mutate(
    Bränsle = case_when(
      Bränsle %in% c("Dieseö", "Diesel") ~ "Diesel",
      Bränsle %in% c("EL", "El") ~ "El",
```

```
    Bränsle %in% c("Miljöbränsle/Hybrid",
                   "Miljöbränsle/hybrid",
                   "Miljöbränsle/ Hybrid",
                   "Miljöbränse/Hybrid",
                   "Hybrid") ~ "Miljöbränsle/Hybrid",
    TRUE ~ Bränsle
  )
 )

## Rechecking the data.

blocket_data %>%
 count(Bränsle, sort = TRUE) %>%
 mutate(
  proportion = n / sum(n)
 )

## And finally checking for missing:

blocket_data %>%
 summarise(missing_bränsle = sum(is.na(Bränsle)))

## None missing!

## Rearranging my bränsle variable so that it compares to the total mean.

blocket_data <- blocket_data %>%
 mutate(Bränsle = as.factor(Bränsle))

contrasts(blocket_data$Bränsle) <- contr.sum
```

```
## This is because my question is on predicting the price of Volvo's, and
## it is more intuitive to compare to the mean price of all Volvo's than to
## a baseline (e.g. Diesel volvos are x cheaper than the mean Volvo).

## Perfect - this is now converted to a factor and now ready for the model.

## Done - so now we have Seller, milage, age, and bränsle. This is 4 variables
## that I would argue are most relevant to predicting sale price on blocket. I
## could be wrong though.

colnames(blocket_data)

## So, I suppose we should really check for the most and least importants. I
## already transformed the 4 most important variables that I believe will influence
## the price, but we need to use all the data we have on hand to make sure.

## So, in our data, we will be considering:

## Växellåda (dummy), drivning (dummy), hästkräfter(numeric), motorstorlek(numeric),
## märke (+ other variables already transformed).

blocket_data %>%
  count(Växellåda, sort = TRUE)

## We can see that växellåda is fine, no strange cases.

blocket_data %>%
  count(Drivning, sort = TRUE)

## We can see that there is some bad info here.
```

```
blocket_data <- blocket_data %>%
 filter(!is.na(Drivning) & Drivning != "SUV") %>%  # Drop NA and SUV
 mutate(
  Drivning = case_when(
   str_to_lower(str_trim(Drivning)) %in% c("tvåhjulsdriven", "tvåhjuldriven") ~ "Tvåhjulsdriven",
   str_to_lower(str_trim(Drivning)) == "fyrhjulsdriven" ~ "Fyrhjulsdriven",
   TRUE ~ Drivning
  ),
  Drivning = factor(Drivning)
 )

blocket_data %>%
 count(Drivning, sort = TRUE)

## That looks better.

str(blocket_data$Hästkrafter)

## Horsepower is in chr, needs to be numeric.

blocket_data <- blocket_data %>%
 mutate(
  Hästkrafter = as.numeric(Hästkrafter)
 )

blocket_data %>%
 count(Hästkrafter, sort = TRUE)

## Motorstorlek and hästkrafter correlation? Nope.

cor(blocket_data$Motorstorlek, blocket_data$Hästkrafter, use = "complete.obs")
```

```
blocket_data %>%
  count(Motorstorlek, sort = TRUE)

hist(blocket_data$Motorstorlek, breaks = 20, col = "skyblue", main = "Histogram av Motorstorlek",
     xlab = "Motorstorlek (liter)")

blocket_data$engine_size_category <- case_when(
  blocket_data$Motorstorlek == 0 ~ "Electric",
  blocket_data$Motorstorlek > 0 & blocket_data$Motorstorlek <= 1899 ~ "Small",
  blocket_data$Motorstorlek >= 1900 & blocket_data$Motorstorlek <= 2299 ~ "Medium",
  blocket_data$Motorstorlek > 2299 ~ "Large",
  TRUE ~ NA_character_
)

blocket_data$engine_size_category <- factor(
  blocket_data$engine_size_category,
  levels = c("Electric", "Small", "Medium", "Large")
)

table(blocket_data$engine_size_category)

## engine_size_category

## Now the question for model.

blocket_data %>%
  count(Modell, sort = TRUE) %>%
  print(n = Inf)

## Replacing incorrectly written models.
```

```
blocket_data$Modell <- blocket_data$Modell %>%
  str_replace_all(regex("^XC 40$", ignore_case = TRUE), "XC40") %>%
  str_replace_all(regex("^XC6$", ignore_case = TRUE), "XC60") %>%
  str_replace_all(regex("^c40$", ignore_case = TRUE), "C40") %>%
  str_replace_all(regex("^V61$", ignore_case = TRUE), "V60") %>%
  str_replace_all(regex("^EC40$", ignore_case = TRUE), "C40")

blocket_data %>%
  count(Modell, sort = TRUE) %>%
  print(n = Inf)

## Here, I decided to group some of the oldest models, as they are all
## at classic era and the amount of categories for model is very large.

classic_models <- c(
  "244", "245", "142", "240 2.1 DL", "244 GL", "850",
  "740", "940", "S70"
)

blocket_data <- blocket_data %>%
  mutate(Modell = if_else(Modell %in% classic_models, "Classic Volvo", Modell))

blocket_data %>%
  count(Modell, sort = TRUE) %>%
  print(n = Inf)

## Think I am happy enough with this.

## Realizing that what I really want to use is the car FEATURES to predict price.
## The car model effectively is what described the features (outside of age/milage
```

```
## for example), so I am thinking that I exclude it?

## So now, I think we are able to build our first regression model?

## Seller (dummy), milage (numeric), age (numeric), bränsle(categorical/effect),
## Växellåda (dummy), drivning (dummy), hästkräfter(numeric), motorstorlek(numeric).

colnames(blocket_data)

## Checking names for the regression.

sapply(blocket_data[, c("Bränsle", "Växellåda", "Drivning")], class)

blocket_data$Växellåda <- as.factor(blocket_data$Växellåda)

## Just converted the last variable to factor.

## Realized I haven't fixed my dependent variable, and a bunch of other variables.

##model <- lm(Price ~ seller_dummy + Bränsle + Växellåda + Drivning +
        ##Hästkrafter + engine_size_category + Age, data = blocket_data)

## Above is what I want to run, so I should check each variable.

str(blocket_data$Price)

unique(blocket_data$Price)

sum(is.na(blocket_data$Price))

## Looks good.
```

```
str(blocket_data$seller_dummy)

unique(blocket_data$seller_dummy)

sum(is.na(blocket_data$seller_dummy))

## Good.

str(blocket_data$Bränsle)

unique(blocket_data$Bränsle)

sum(is.na(blocket_data$Bränsle))

# Good.

str(blocket_data$Växellåda)

unique(blocket_data$Växellåda)

sum(is.na(blocket_data$Växellåda))

## Good.

str(blocket_data$Drivning)

unique(blocket_data$Drivning)

sum(is.na(blocket_data$Drivning))
```

```
## Good.

str(blocket_data$Hästkrafter)

unique(blocket_data$Hästkrafter)

sum(is.na(blocket_data$Hästkrafter))

## Good.

str(blocket_data$Motorstorlek)

unique(blocket_data$Motorstorlek)

sum(is.na(blocket_data$Motorstorlek))

## Cleaning:

##blocket_data$Motorstorlek <- trimws(blocket_data$Motorstorlek)
##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "N/A"] <- NA

##blocket_data$Motorstorlek <- as.numeric(blocket_data$Motorstorlek)

##unique(blocket_data$Motorstorlek)
##sum(is.na(blocket_data$Motorstorlek))

## Checking to see if N/A's line up for electric.

##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "211"] <- "2110"

##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "2521 "] <- "2521"
```

```
##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "1969 "] <- "1969"

##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "2400 "] <- "2400"

##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "1498 "] <- "1498"

##blocket_data$Motorstorlek[blocket_data$Motorstorlek == "1984 "] <- "1984"

##unique_values <- unique(blocket_data$Motorstorlek)

##non_numeric_values <- unique_values[is.na(as.numeric(unique_values))]

##non_numeric_values

##blocket_data$Motorstorlek <- trimws(blocket_data$Motorstorlek)

##non_numeric_values <-
unique(blocket_data$Motorstorlek[is.na(as.numeric(blocket_data$Motorstorlek))])

##non_numeric_values

## Did a lot of experimenting here to fix the motorstorlek variable.
## Ended up converting it in excel to kill the deadspace.
## New variable is Enginesize.

str(blocket_data$Enginesize)

unique(blocket_data$Enginesize)

sum(is.na(blocket_data$Enginesize))
```

```
## Issue persists? I guess we do it the old fashioned way - manually cleaned
## in excel.

table(is.na(blocket_data$Enginesize), blocket_data$Bränsle)

## Perfect. Now, this should be converted to numerical.

blocket_data$Enginesize <- as.numeric(blocket_data$Enginesize)

sum(is.na(blocket_data$Enginesize))

## Looks better than before, but 2 additional N/A's were introduced unexpectedly.

## Did a final check, should be fixed in the dataset now.

## Model should be ready to run now.

levels(blocket_data$Bränsle)

summary(blocket_data[, c("Bränsle", "Växellåda", "Drivning", "engine_size_category")])

model <- lm(Price ~ ., data = blocket_data %>%
            select(Price, seller_dummy, Milage, Age, Bränsle, Växellåda, Drivning, Hästkrafter,
engine_size_category, Modell))
vif(model)

## Checking which variables to log transform.

numeric_data <- blocket_data %>%
  select(where(is.numeric)) %>%
```

```
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")



ggplot(numeric_data, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "#2C3E50", color = "white") +
  facet_wrap(~ Variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Histograms of Numeric Variables")

## transforming age, hästkrafter, milage, and price.

blocket_data <- blocket_data %>%
  mutate(
    log_Price = log(Price),
    log_Age = log(Age + 1),
    log_Milage = log(Milage + 1),
    log_Hästkrafter = log(Hästkrafter + 1)
  )

## Recheck

log_data_long <- blocket_data %>%
  select(log_Price, log_Age, log_Milage, log_Hästkrafter) %>%
  pivot_longer(everything(), names_to = "Variable", values_to = "Value")

ggplot(log_data_long, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Histograms of Log-Transformed Variables")
```

```
## Log milage and log price have become left skewed. Will try a sqrt transform.

blocket_data <- blocket_data %>%
  mutate(
    sqrt_Price = sqrt(Price),
    sqrt_Milage = sqrt(Milage + 1)
  )

## Checking dist again.

ggplot(blocket_data, aes(x = sqrt_Price)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  labs(title = "Histogram of sqrt(Price)", x = "sqrt(Price)", y = "Count")

ggplot(blocket_data, aes(x = sqrt_Milage)) +
  geom_histogram(bins = 30, fill = "seagreen", color = "white") +
  labs(title = "Histogram of sqrt(Milage)", x = "sqrt(Milage)", y = "Count")

## Damn, they're some nice distributions!

##model <- lm(Price ~ ., data = blocket_data %>%
          ##select(Price, seller_dummy, Milage, Age, Bränsle, Växellåda, Drivning, Hästkrafter,
engine_size_category, Modell))

## Final check on the variables

selected_vars <- blocket_data %>%
  select(sqrt_Price, sqrt_Milage, log_Age, log_Hästkrafter, seller_dummy, Växellåda, Bränsle,
Drivning, engine_size_category)

sapply(selected_vars, class)
```

```
## seller_dummy is numeric - wrong.

blocket_data$seller_dummy <- as.factor(blocket_data$seller_dummy)

## REGRESSION MODEL 1:

levels(blocket_data$Bränsle)
levels(blocket_data$Växellåda)
levels(blocket_data$engine_size_category)

blocket_data$engine_size_category <- relevel(blocket_data$engine_size_category, ref = "Electric")

blocket_data$Bränsle <- relevel(blocket_data$Bränsle, ref = "El")

levels(blocket_data$Bränsle)
levels(blocket_data$engine_size_category)

## Reorganizing the order of the categories.

blocket_data$Bränsle <- factor(blocket_data$Bränsle,
                               levels = c("El", "Miljöbränsle/Hybrid", "Diesel", "Bensin"))

blocket_data$engine_size_category <- factor(blocket_data$engine_size_category,
                                            levels = c("Electric", "Large", "Medium", "Small"))

model <- lm(sqrt_Price ~ engine_size_category,
            data = blocket_data)

summary(model)

##vif(model)
```

```
## Getting really dug in with managing the issue of electric cars only being
## new vs diesel/bensin being mainly older. Considering cohorting age.

blocket_data$age_cohort <- cut(
  blocket_data$Modellår,
  breaks = c(0, 1999, 2009, 2014, 2019, Inf),
  labels = c("<2000", "2000–2009", "2010–2014", "2015–2019", "2020+")
)

blocket_data$age_cohort <- factor(
  blocket_data$age_cohort,
  levels = c("2020+", "2015–2019", "2010–2014", "2000–2009", "<2000")
)

table(blocket_data$age_cohort)

model <- lm(sqrt_Price ~ age_cohort,
          data = blocket_data)

summary(model)

## All of this is very interesting. Age/age cohorts are by far the strongest
## predictor of price for listings on blocket for Volvo's, to the point
## that it is explaining upwards of 80% of the variation in price.

## I will attempt a best subset selection, but I expect a smaller model with
## 2-5 variables with the primary explanatory variable of age/age cohorts will
## be the best model.

best_subset <- regsubsets(
```

```
  sqrt_Price ~ age_cohort + Bränsle + log_Age + seller_dummy
  + Växellåda + sqrt_Milage + Drivning + log_Hästkrafter + engine_size_category,
  data = blocket_data,
  nvmax = 9
)

summary_best <- summary(best_subset)

summary_best$adjr2
summary_best$bic

plot(summary_best$adjr2, type = "b", xlab = "Number of Predictors", ylab = "Adjusted R²")

## 7 predictors is apparently the best - but at the same time, the R2 is already
## very high, even after a single predictor.

# Find the number of predictors with the highest adjusted R²
best_model_index <- which.max(summary_best$adjr2)

# Alternatively, for the lowest BIC:
# best_model_index <- which.min(summary_best$bic)

# Get the model with the best number of predictors
best_model <- summary_best$which[best_model_index, ]
best_model

## This gives some useful information for what is giving power to the model,
## but at the same time should only be really taken with a pinch of salt.

## The key takeaways are that the selection suggests that bränsle is not as
## important for the model performance as initially believed, and that transmission
```

```
## the wheel drive and hoursepower are more powerful, but that all of these pale

## in comparison to our age cohorts. Interestingly, log_Age is also included as valuable

## even with our age cohorts.


## From this, I think I can deduce that age_cohort, växellåda, sqrt_Milage, drivning,

## log_hästkrafter are all valuable for the model. Let's look at it:


lm_model <- lm(sqrt_Price ~ age_cohort + Växellåda + sqrt_Milage + Drivning + log_Hästkrafter, data
= blocket_data)


summary(lm_model)


## Good model?? Looks nice.


## Trying a quick lasso


x <- model.matrix(sqrt_Price ~ age_cohort + Växellåda + sqrt_Milage + Drivning + log_Hästkrafter,
data = blocket_data)[,-1]

y <- blocket_data$sqrt_Price


# Fit a lasso model (alpha = 1 for lasso)

lasso_model <- glmnet(x, y, alpha = 1)


# Plot the lasso path

plot(lasso_model, xvar = "lambda")


# Cross-validation to choose the optimal lambda

cv_lasso <- cv.glmnet(x, y, alpha = 1)

plot(cv_lasso)


# Get the optimal lambda

cv_lasso$lambda.min
```

```
# Fit the lasso model with the optimal lambda
lasso_optimal <- glmnet(x, y, alpha = 1, lambda = cv_lasso$lambda.min)

# Coefficients of the lasso model
coef(lasso_optimal)

####################

# Linear model R-squared
summary(lm_model)$adj.r.squared

# Lasso model R-squared
lasso_pred <- predict(lasso_optimal, s = cv_lasso$lambda.min, newx = x)
lasso_r2 <- 1 - sum((y - lasso_pred)^2) / sum((y - mean(y))^2)
lasso_r2

####################

# Calculate RMSE for the linear regression model
lm_pred <- predict(lm_model, newdata = blocket_data)
lm_rmse <- sqrt(mean((blocket_data$sqrt_Price - lm_pred)^2))
lm_rmse

# Calculate RMSE for the lasso regression model
lasso_rmse <- sqrt(mean((y - lasso_pred)^2))
lasso_rmse

######################

# AIC for linear regression model
```

```
AIC(lm_model)

# AIC for lasso regression model
# To calculate AIC for lasso, we can use the residual sum of squares and number of predictors
lasso_rss <- sum((y - lasso_pred)^2)
n <- length(y)  # number of observations
k <- sum(coef(lasso_optimal) != 0)  # number of non-zero coefficients
lasso_aic <- n * log(lasso_rss / n) + 2 * k
lasso_aic

########################

## To conclude here - the lasso model performs better and is also likely avoiding
## any overfitting by shrinking unimportant coefficients to 0. This means that
## it will likely perform somewhat better on unseen data, even if it was trained
## on the exact same predictors.

########################

## Do I need to test this on a test set then?

## saveRDS(lasso_optimal, file = "lasso_model.rds")

##lasso_loaded <- readRDS("lasso_model.rds")blocket_data %>%

## cut code to build coefficients so below code won't work now, but did prior.

# Convert lasso coefficients (sparse matrix) to a data frame
lasso_table <- data.frame(
  term = rownames(lasso_coefs),
  estimate = as.numeric(lasso_coefs)
```

```
)

# View it
print(lasso_table)

clean_names <- c(
  "(Intercept)" = "(Intercept)",
  "age_cohort2015–2019" = "Ålder: 2015–2019",
  "age_cohort2010–2014" = "Ålder: 2010–2014",
  "age_cohort2000–2009" = "Ålder: 2000–2009",
  "age_cohort<2000"     = "Ålder: <2000",
  "VäxellådaManuell"    = "Växellåda: Manuell",
  "sqrt_Milage"         = "Miltal (roten ur)",
  "DrivningTvåhjulsdriven" = "Drivning: Tvåhjulsdrift",
  "log_Hästkrafter"     = "Hästkrafter (log)"
)

lasso_table_clean <- lasso_table %>%
  mutate(term = clean_names[term])

# View
print(lasso_table_clean)

lasso_table_clean %>%
  gt() %>%
  fmt_number(columns = estimate, decimals = 2) %>%
  tab_header(title = "Lasso Regression Coefficients (Cleaned)")

explanations <- c(
  "(Intercept)" = "Baseline price (2020+ model, automatic, 0 mileage, FWD, 0 horsepower",
  "Ålder: 2015–2019" = "2015–2019 models cost ~43k SEK less than 2020+",
```

```
  "Ålder: 2010–2014" = "2010–2014 models cost ~123k SEK less than 2020+",
  "Ålder: 2000–2009" = "2000–2009 models cost ~212k SEK less than 2020+",
  "Ålder: <2000"     = "Older than 2000 cost ~155k SEK less than 2020+",
  "Växellåda: Manuell" = "Manual cars are ~30k SEK cheaper",
  "Miltal (roten ur)" = "Higher mileage reduces price (complex root to log comparison)",
  "Drivning: Tvåhjulsdrift" = "Two-wheel drive cars are ~41k SEK cheaper",
  "Hästkrafter (log)" = "1% more horsepower increases price by ~990 SEK"
)

# Add the explanation column
lasso_table_final <- lasso_table_clean %>%
  mutate(Explanation = explanations[term])

# View
print(lasso_table_final)

lasso_table_final %>%
  gt() %>%
  fmt_number(columns = estimate, decimals = 2) %>%
  cols_label(
    term = "Feature",
    estimate = "Estimate (in Thousand SEK)",
    Explanation = "Interpretation"
  ) %>%
  tab_header(title = "Lasso Regression Coefficients and Interpretations")

percentage_differences <- c(
  21.44, -38.30, 13.30, 12.87, -2.15, 2.55, 33.20, 18.32, -4.00, -5.32,
  16.04, 6.57, 9.76, -12.43, -9.06, -0.65, 21.37, 6.64, -35.92, -113.62,
  -48.79, -74.30, 34.60, 1.49, 18.43, 48.18, 3.75, -4.19
)
```

```
# Make QQ plot
qqnorm(percentage_differences)
qqline(percentage_differences, col = "blue")
```

Källförteckning

Kadhammar, A. & Wong, W. (2021) What in the ad affects how fast a car is sold on Blocket. Accessed from: https://www.diva-portal.org/smash/get/diva2:1563308/FULLTEXT01.pdf [Date Accessed: 24-04-2025]

Kumar, S. & Sinha, A. (2024) Predicting Used Car Prices with Regression Techniques. International Journal of Computer Trends and Technology. Vol. 72 (6) pp. 132-141. Accessed from: https://www.ijcttjournal.org/2024/Volume-72%20Issue-6/IJCTT-V72I6P118.pdf [Date Accessed: 24-04-2025]

Statistiska centralbyrån (SCB) (2025) SCB Open Data API. Accessed from: https://www.scb.se/vara-tjanster/oppna-data/api-for-oppna-data/ [Date Accessed: 24-04-2025]