Prediktion av begagnade bilpriser med hjälp av regressionsanalys

Kawser Ayoub

EC Utbildning

R-programmering

2024-04

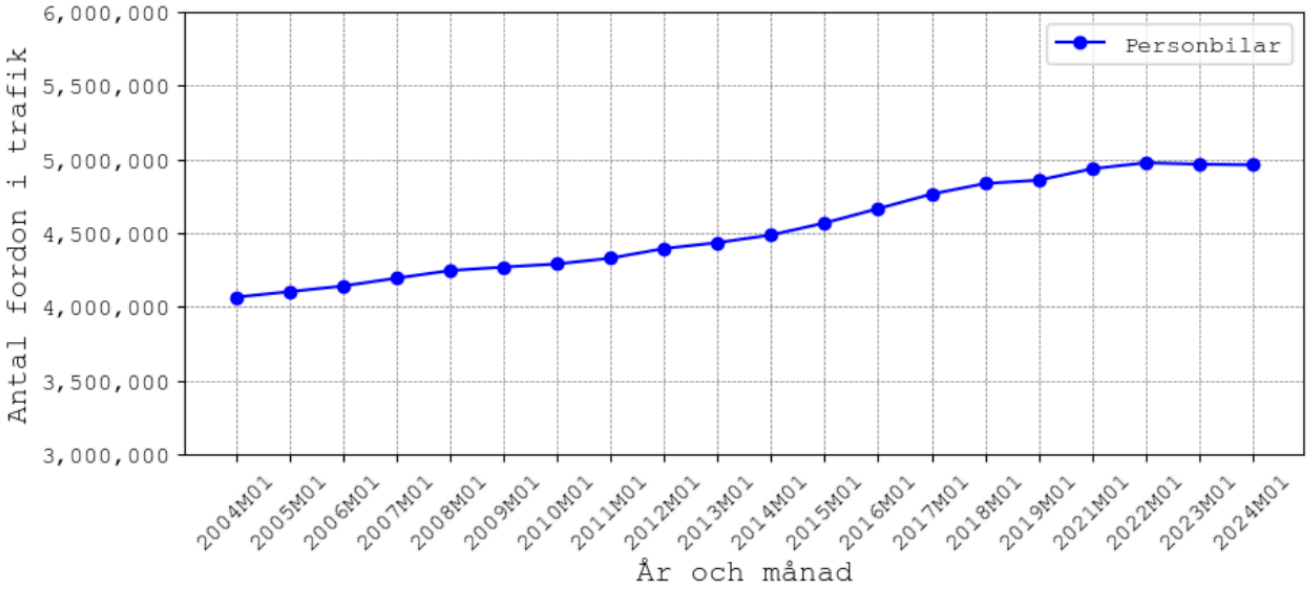
Abstract

This report describes an analysis of car sales data, focusing on exploring and modeling the prices of used cars based on their characteristics. The analysis involved data cleansing, handling errors and exploratory analysis to understand the distribution of the data. Furthermore, Lasso and Ridge was used to create a model that predicts prices with high precision. Model performance was evaluated using Root Mean Square (RMSE) and Mean Absolute Percentage Error (MAPE), with Lasso regression showing the best results out of all the models.

Innehållsförteckning

1. Inledning

Den stadiga ökningen av antal fordon i trafik indikerar en växande marknad, vilket tyder på en större efterfrågan på fordon över tid. Med tanke på detta sammanhang kan det vara ett värdefullt tillvägagångssätt att utveckla en automatiserad pris prediktions modell.



Figur 1. Antal fordon i trafik mellan 2004-2024. (API)

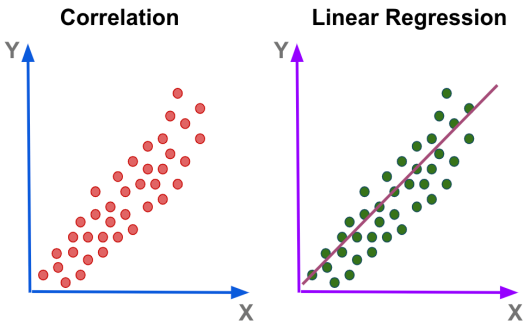
Syftet med denna rapport är att bygga modeller som kan hjälpa till att förutsäga priser. För att uppfylla syftet kommer följande frågeställningar att besvaras:

1. Hur väl predikterar modellen bilpriser?
2. Vilka variabler påverkar bilpriser mest?
3. Hur utvärderas modellernas prestanda?

2. Teori

2.1 Linjär regression

Linjär regression en statistisk metod som används för att förutsäga värdet av en beroende variabel baserat på värdet/värdena för en eller flera oberoende variabler. Metoden går på att anpassa en rät linje till data som minimerar avvikelserna mellan observerade värden och värden som förutspås av modellen. Detta görs vanligtvis med metoden “minsta kvadrater”. (IBM)



Figur 2. Linjär regression

2.2 Lasso regression

Lasso regression eller även så kallad L1 regularization minskar överanpassning i modeller genom att lägga till ett straff som motsvarar koefficientens absoluta värde. Detta tillvägagångssätt hjälper inte bara till att minimera den kvarvarande summan av kvadrater utan också i feature selection genom att krympa mindre viktiga koefficienter till noll. Det är särskilt effektivt i hög dimensionella data scenarier där det förbättrar modellens enkelhet och tolkningsbarhet genom att endast behålla de mest relevanta prediktorerna. (IBM)

2.3 Ridge regression

Ridge regression eller L2 regulasering, adresserar multikollinearitet och överanpassning i regressionsmodeller genom att lägga till en straff term till förlustfunktionen. Detta straff är proportionellt mot kvadraten på koefficientens storlek, vilket minskar effekten av högt korrelerade prediktorer. Ridge eliminerar inte prediktorer helt som Lasso gör men den minskar koefficienterna proportionellt, vilket hjälper till att stabilisera modell förutsägelserna över olika datamängder. det är särskilt användbart i scenarier med många prediktorer eller feature korrelation. (IBM)

2.4 Multikollinearitet

Multikollinearitet i regressionsanalys avser frågan om oberoende variabler är starkt korrelerade, vilket komplicerar tolkningen av modellens koefficienter. Det diagnostiseras med hjälp av Variance Inflation Factor (VIF), med värden över 5 som tyder på allvarlig multikollinearitet. Lösningar inkluderar att ta bort korrelerade variabler eller tillämpa realiserings tekniker som Ridge-regression (statisticsbyjim.com).

2.6 RMSE och MAPE

Root Mean Square Error (RMSE) mäter genomsnittliga avvikelsen för predikterade värden från observerade värden i en modell, och kvantifierar prediktionsfel. Den beräknar kvadratroten av de genomsnittligt kvadrerade skillnaderna mellan förutsagda och faktiska utfall, vilket gör den känslig för extremvärden. RMSE värden uttrycks i samma enheter som data, vilket ger en tydlig fel storlek. (statisticsbyjim.com)

Mean Absolute Percentage Error (MAPE) kvantifiera forecasting accuracy genom att uttrycka den genomsnittliga absoluta procentuella skillnaden mellan förutsagda och faktiska värden. Den beräknas genom att genomsnittet av de absoluta procentuella felen över datapunkter, vilket effektivt skalar fel till procentuella termer för enklare förståelse. MAPE är mest effektivt med dataset som inte innehåller extrema värden eller nollor. (statisticshowto.com)

3. Metod

3.1 Datainsamling

Jag var i samma grupp som Abdulrahman, Alia , Anton, Daniel, George, Goran, Jesper och John. Vi enades snabbt in om att använda web scraping som metod för datainsamlingen eftersom en av gruppmedlemmarna introducerade det för oss och delade en fil som han hade extraherat innan grupperna bildades. Efter att ha granskat den initiala webbscrapingen insåg vi att vi behövde avgränsa oss och välja variabler som skulle ge oss ett mer koncentrerat resultat. Som nästa steg genomfördes ny web scraping som vi kunde enas kring. Utöver webscraping samlade vi även in data om 10-30 bilar manuellt för att skapa en POC för webbscrapingen. Bra med grupparbetet var att vi snabbt kunde enas om en metod för datainsamling. Vad som skulle kunna utvecklas är att vi inte ska fokusera allt för mycket på detaljer och se saker från ett bredare perspektiv.

3.2 Arbetsflöde

Datainsamlingen gjordes genom en web-scraping teknik, riktad mot begagnade bilar på blocket. Urvalskriterier definierades för att effektivisera datasetet och fokusera analysen på relevanta faktorer. Specifikt var datamängden begränsad till biltyper som halvkombi, kombi, SUV och sedan. Endast bilar tillverkade mellan 2014 och 2024 inkluderades, och de valda märkena var Kia, Volkswagen, Volvo, Audi, BMW, Peugeot, Opel och Toyota. Prisintervallet sattes med en lägsta tröskel på 20 000 kr. Båda privata och företagssäljare valdes och fordon omfattade manuella och automatiska växellådor, som drivs på antingen bensin eller diesel. Det samlades 7107 bilar.

Arbetsflödet gör data analysen omfattar datarensning, inklusive borttagning av felaktiga inmatningar, hantering av saknade värden och eliminering av dubbletter. Den rensade datan omvandlas sedan för analys; till exempel omvandlas kategoriska variabler till faktorer och kontinuerliga variabler normaliseras med logaritmiska transformationer. Sedan byggs analytiska modeller för att utforska relationer inom data, med hjälp av tekniker som linjär regression och regularisering metoder som Lasso och Ridge regression.

Datasetet är uppdelad i tränings, validerings och test delar för att säkerställa modellens robusthet och för att utvärdera prestandamått som RMSE och MAPE. Genom hela analysen används olika datavisualisering och diagnostiska verktyg för att bedöma modell validitet och påverkan på prediktorer. Det skapades även en API för att hämta data från SCB om antal fordon i trafik. Det gjordes i Jupyter Notebook med Python och det extraherades information om bilar mellan 2004 och 2024.

4. Resultat och Diskussion

Flera regressionsmodeller är konstruerade för att förutsäga bilpriser. Det skapades sammanlagt 5 modeller. Initiala modeller använder vanliga och log transformerade resultat för att mäta olika aspekter av data relationerna. Modeller visar rimliga fits med R-squared värden som förbättrar hanteringen efter avvikelser och datarensning t.ex. från 0,6312 i den första modellen till 0,7378 i den tredje modellen efter att ha tagit bort högs hävstångs punkter. Att använda log transformerade price och miles förbättrade modellen, vilket kan ses med lägre residualer och högre justerade R-square värden.

Från de linjära regression modellernas resultat kan vi se att de mest inflytelserika variablerna är Brand, Model Year, Engine och Gears. Vissa märken som BMW och Audi har positiva koefficienter som indikerar högre pris medan andra märken som Kia och Opel har negativa koefficienter som indikerar ett lägre pris jämfört med basnivån. Det finns en positiv relation mellan bildens årsmodell och pris, vilket betyder att nyare bilar tenderar att vara dyrare. Dieselbilar har en tendens att kosta mer än bensinbilar. Bilar med manuell växellåda har en negativ koefficient, vilket tyder på att de generellt sett är billigare än bilar med automatiska växellåda.

Lasso och Ridges regularisering tekniker används för att förbättra modellens generalisering och förhindra överanpassning. Det användes korsvalidering för att välja optimala lambdavärden för båda metoderna, vilket säkerställer att modellerna varken är för komplexa eller för enkla. Modellerna utvärderas med hjälp av RMSE och MAPE för validering och test datan. Lasso (modell 5) överträffar andra med lägsta RMSE och MAPE, vilket indikerar högre noggrannhet och bättre prestanda för att förutsäga log transformerade priser.

Model	RMSE	MAPE
Model 1	2.412653e+05	1.879594e+06
Model 2	2.065362e-01	1.328278e+00
Model 3	2.078631e-01	1.326794e+00
Model 4	5.028155e-02	3.197687e-01
Model 5	1.081455e-02	7.094230e-02

Denna överlägsenhet bekräftades genom slutlig testning och jämförelse av faktiska kontra förutspådda priser, vilket visar Lasso modellens precision i praktiska sammanhang.

	Actual_Price	s1
1	214900	214973.3
2	219900	219827.6
3	199000	199514.4
4	149800	151435.8
5	189900	190650.9
6	129000	130978.2
7	349800	344985.0
8	149000	150650.6
9	234800	234274.5
10	176000	177087.9

5. Slutsatser

Analysen visade potentialen hos automatiserade prediktiva modeller för att tolka marknads dynamiken. Modellernas effektivitet när det gäller att förutsäga bilpriser utvärderades med hjälp av en robust metod, främst med fokus på linjär regression och dess förlängningar, Lasso och Ridge regressioner. Prestandan kvantifierades med hjälp av mätvärden: RMSE och MAPE, med resultat som indikerar att modellen Lasso uppnådde en hög noggrannhet i att förutsäga priser. Denna modell minimerade förutsägelse fel och var även den mest tillförlitliga när den testades mot osynlig data, vilket bekräftar dess användbarhet i praktiska scenarier. Sammanfattningsvis tog utvecklingen av dessa prediktiva modeller upp nyckelfrågor om de faktorer som mest signifikant påverkar bil priserna och modellernas prediktiva noggrannhet.

6. Teoretiska frågor

1. Beskriv kortfattat vad en Quantile Quantile (QQ) plot är.

En Quantile-Quantile plot är ett grafiskt verktyg som används för att jämföra kvantilerna för två sannolikhetsfördelningar. Genom att plotta kvantilerna för en data set mot en annan hjälpder att avgöra om de kommer från liknande distributioner eller bedöma hur ett data set passar en teoretisk fördelning.

2. Din kollega Karin frågar dig följande: ”Jag har hört att i Maskininlärning så är fokus på prediktioner medan man i statistisk regressionsanalys kan göra såväl prediktioner som statistisk inferens. Vad menas med det, kan du ge några exempel?” Vad svarar du Karin?

Det stämmer. Inom maskininlärning ligger fokus på att göra korrekta förutsägelser med hjälp av algoritmer som lär sig av data. Däremot tillåter statistisk regressionsanalys både förutsägelser och statistisk slutledning och utforskar sambanden mellan variabler. T.ex. kan linjär regression användas för att förutsäga och utvärdera hur husstorleken påverkar priserna.

3. Vad är skillnaden på ”konfidensintervall” och ”prediktionsintervall” för predikterade värden?

Ett konfidensintervall uppskattar osäkerheten kring en populations parameter, som ett medelvärde. Däremot uppskattar ett prediktionsintervall intervallet där framtida observationer sannolikt kommer att falla, vilket tar hänsyn till individuella data punkts variationer, vilket gör det bredare än ett kondifensintervall

4. Den multipla linjära regressionsmodellen kan skrivas som: $Y\,\beta 0\,\beta 1\,x1\,\beta 1\,x2\,\beta pxp\,\varepsilon$. Hur tolkas beta parametrarna?

I en multipel linjär regressionsmodell representerar beta koefficienterna den förväntande förändringen i den beroende varaibeln(Y) för en ökning på en enhet i den respektive oberoende variablerna (x1,x2…,xp), vilket håller alla andra variabler konstanta.

5. Din kollega Hassan frågar dig följande: ”Stämmer det att man i statistisk regressionsmodellering inte behöver använda träning, validering och test set om man nyttjar mått såsom BIC Vad är logiken bakom detta?” Vad svarar du Hassan?

Det stämmer. BIC hjälper till att utvärdera modellens prestanda och komplexitet direkt genom att straffa överanpassning, vilket tjänar ett liknande syfte som att använda separata data undergrupper.

6. Förklara algoritmen nedan för ”Best subset selection”.

”Best subset selection” är en algoritm i statistik som identifierar den mest prediktiva delmängder av variabler för en regressionsmodell. Det fungerar genom att utvärdera alla möjliga kombinationer av prediktorvariabler, anpassa en modell för varje kombination och sedan välja den delmängd som bäst förbättrar modellen enligt ett valt kriterium, såsom den lägsta residual sum of squares eller det högsta justerade R-squared värdet. Detta sättet säkerställer valet av den optimala uppsättningen av prediktorer men kan vara beräknings intensiv för ett stort antal prediktorer.

1. Börja med en grundläggande modell M0 som beräknar medelvärdet för varje observation.
2. För varje räkning av förklarande variabler (1 till p), beräkna alla möjliga kombinationer, välj den bästa modellen per räkning med RSS och R^2.
3. Välj den bästa övergripande modellen från M0 till Mp med hälp av valideringstekniker som AIC, BIC eller justerad R^2.

7. Ett citat från statistikern George Box är: “All models are wrong, some are useful.” Förklara vad som menas med det citatet.

Jag tolkar det som att även om ingen statistisk modell kan fånga alla aspekter av vår verklighet på ett perfekt sätt (på grund av förenkling och antagande) så kan vissa modeller fortfarande vara värdefulla eftersom de ger användbara approximationer eller insikter som kan informera och vägleda.

Appendix

```r
#Load libraries
library(readxl)
library(skimr)
library(dplyr)
library(car)
library(Metrics)
library(leaps)
library(MASS)
library(glmnet)

#Load data
data <- read_excel("C:\\cars_data.xlsx")
View(data)

#EDA--------------------------------------------------------------------------
dim(data)
skim(data)
summary(data)

#Examine unique values
table(data$Brand)
table(data$Engine)
table(data$Gears)
table(data$Region)

#Remove incorrect values in 'Brand'
data <- data[!data$Brand %in% c("2016", "2021", "VW", "XC60", "Söker"), ]
table(data$Brand)

#Handle missing values
sum(is.na(data)) #645
sapply(data, function(x) sum(is.na(x)))
data$Dealer[is.na(data$Dealer)] <- "Private Dealers"
sapply(data, function(x) sum(is.na(x)))
data$Dealer[data$Dealer != "Private Dealers"] <- "Corporate Dealers"
table(data$Dealer)
```

```r
#Remove duplicate rows
sum(duplicated(data)) #85
data <- data[!duplicated(data), ]
sum(duplicated(data))

#Convert columns to appropriate data types
data <- data %>%
  mutate_at(c("ModelYear", "Miles", "Price"), as.numeric) %>%
  mutate_at(c("Brand", "Model", "Engine", "Gears", "Region", "Dealer"), as.factor)

str(data)

#Drop columns with high variability in values
data <- data[, !(names(data) %in% c("Model", "Region"))]

#Handle outliers
numerical_columns <- c('ModelYear', 'Price', 'Miles')

remove_outliers <- function(data, columns) {
  for (col in columns) {
    q1 <- quantile(data[[col]], 0.25)
    q3 <- quantile(data[[col]], 0.75)
    iqr <- q3 - q1
    lower_bound <- q1 - 1.5 * iqr
    upper_bound <- q3 + 1.5 * iqr
    data <- data[data[[col]] >= lower_bound & data[[col]] <= upper_bound, ]
  }
  return(data)
}

clean_data <- remove_outliers(data, numerical_columns)

#Add log-transformed columns for 'Price' and 'Miles'
clean_data$LogPrice <- log(clean_data$Price)
clean_data$LogMiles <- log(clean_data$Miles)

summary(clean_data)
skim(clean_data)
```

```r
#View relationships among numerical variables
correlation_matrix <- cor(clean_data[, sapply(clean_data, is.numeric)])
print(correlation_matrix)

#Model Building-------------------------------------------------------------
#Split data
spec = c(train = .6, validate = .2, test = .2)
set.seed(123)
g = sample(cut(
  seq(nrow(clean_data)),
      nrow(clean_data)*cumsum(c(0,spec)),
      labels = names(spec)
))

res = split(clean_data, g)

train <- res$train    #Training set
val <- res$validate   #Validation set
test <- res$test      #Test set

#Model 1
model_1 <- lm(Price ~ . -LogPrice -LogMiles, train)
summary(model_1)
par(mfrow=c(2,2))
plot(model_1)
vif(model_1)

#Model 2
model_2 <- lm(LogPrice ~ . -Price -Miles, train)
summary(model_2)
par(mfrow=c(2,2))
plot(model_2)
vif(model_2)
```

```r
#Model 3
#Detect and remove high leverage points using Cook's Distance
cook_threshold <- 4 /nrow(train)
high_leverage <- which(cooks.distance(model_2) > cook_threshold)
clean_train <- train[-high_leverage,]

model_3 <- lm(LogPrice ~ . -Price -Miles, clean_train)
summary(model_3)
par(mfrow = c(2, 2))
plot(model_3)
vif(model_3)

#Lasso and Ridge
x_train <- model.matrix(~ . -Price -Miles, clean_train)[, -1]
y_train <- clean_train$LogPrice

# Model 4
ridge_cv <- cv.glmnet(x_train, y_train, alpha = 0)
ridge_lambda <- ridge_cv$lambda.min
model_4 <- glmnet(x_train, y_train, alpha = 0, lambda = ridge_lambda)

# Model 5
lasso_cv <- cv.glmnet(x_train, y_train, alpha = 1)
lasso_lambda <- lasso_cv$lambda.min
model_5 <- glmnet(x_train, y_train, alpha = 1, lambda = lasso_lambda)
```

```r
# Model evaluation on validation set-------------------------------------------
calculate_mape <- function(actual, predicted) {
  mean(abs((actual - predicted) / actual)) * 100  #Calculate MAPE
}

x_val <- model.matrix(~ . -Price -Miles, val)[, -1]
y_val <- val$LogPrice

models <- list(model_1, model_2, model_3, model_4, model_5)
predictions <- list(
  predict(model_1, newdata = val),
  predict(model_2, newdata = val),
  predict(model_3, newdata = val),
  predict(model_4, s = ridge_lambda, newx = x_val),
  predict(model_5, s = lasso_lambda, newx = x_val)
)
results_val <- data.frame(
  Model = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5"),
  RMSE = sapply(predictions, function(pred) rmse(val$LogPrice, pred)),
  MAPE = sapply(predictions, function(pred) calculate_mape(val$LogPrice, pred))
)

results_val

#Predictions on test data (best model)
x_test <- model.matrix(~ . -Price -Miles, test)[, -1]
y_test <- test$LogPrice

test_pred <- predict(model_5, s = lasso_lambda, newx = x_test)

results_test <- data.frame(
  RMSE_test = c(rmse(test$LogPrice, test_pred)),
  MAPE_test = c(calculate_mape(test$LogPrice, test_pred))
)

results_test

#Compare the predictated value and the actual value
test_pred <- predict(model_5, s = lasso_lambda, newx = x_test)

comparison <- data.frame(
  Actual_Price = exp(y_test),
  Predicted_Price = exp(test_pred)
)

print("First 10 rows:")
print(head(comparison, 10))

print("Last 10 rows:")
print(tail(comparison, 10))
```

Källförteckning

https://www.ibm.com/topics/linear-regression - hämtad 29-4-2024

https://www.ibm.com/topics/ridge-regression - hämtad 29-04-2024

https://www.ibm.com/topics/lasso-regression - hämtad 29-02-2024

https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/ - hämtad 29-02-2024

https://statisticsbyjim.com/regression/root-mean-square-error-rmse/ - hämtad 29-02-2024

https://www.statisticshowto.com/mean-absolute-percentage-error-mape/ - hämtad 29-02-2024