Linear Regression Model, Predictions, Accuracy

Devan Morehouse

2023-09-20

Linear Regression Model

Loading the data

```
car.df <- read.csv("C:\\Users\\dev46\\OneDrive\\Desktop\\School Documents\\Spring 2023\\MGQ 408 Bus. Analytics & Data Scie
nce\\Data\\ToyotaCorolla.csv")

# use first 1000 rows of data
car.df <- car.df[1:1000, ]
# select variables for regression
selected.var <- c(3, 4, 7, 8, 9, 10, 12, 13, 14, 17, 18)
```

Partition data

```
set.seed(1)  # set seed for reproducing the partition
train.index <- sample(c(1:1000), 600)
train.df <- car.df[train.index, selected.var]
valid.df <- car.df[-train.index, selected.var]
```

Training the model

```
# use lm() to run a linear regression of Price on all 11 predictors in the
# training set.
# use . after ~ to include all the remaining columns in train.df as predictors.
car.lm <- lm(Price ~ ., data = train.df)

#  use options() to ensure numbers are not displayed in scientific notation.
options(scipen = 999)
summary(car.lm)
```

```
##
## Call:
## lm(formula = Price ~ ., data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9781.2  -729.9     0.9   739.3  6912.9
##
## Coefficients:
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     -4754.379821  1661.719608  -2.861             0.004372 **
## Age_08_04        -133.271592     4.901960 -27.187 < 0.0000000000000002 ***
## KM                 -0.020992     0.002304  -9.111 < 0.0000000000000002 ***
## Fuel_TypeDiesel   896.206322   603.164063   1.486             0.137857
## Fuel_TypePetrol  2191.368250   575.629429   3.807             0.000155 ***
## HP                 37.257956     5.233283   7.119     0.00000000000317 ***
## Met_Color          51.315188   123.395390   0.416             0.677664
## Automatic          63.567598   262.282017   0.242             0.808583
## CC                  0.010747     0.097711   0.110             0.912456
## Doors             -55.700492    63.966255  -0.871             0.384230
## Quarterly_Tax      13.080021     2.608396   5.015     0.00000070465597 ***
## Weight             16.219638     1.526915  10.622 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1392 on 588 degrees of freedom
## Multiple R-squared:  0.8703, Adjusted R-squared:  0.8679
## F-statistic: 358.7 on 11 and 588 DF,  p-value: < 0.00000000000000022
```

Making predictions on a new set and testing the accuracy

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

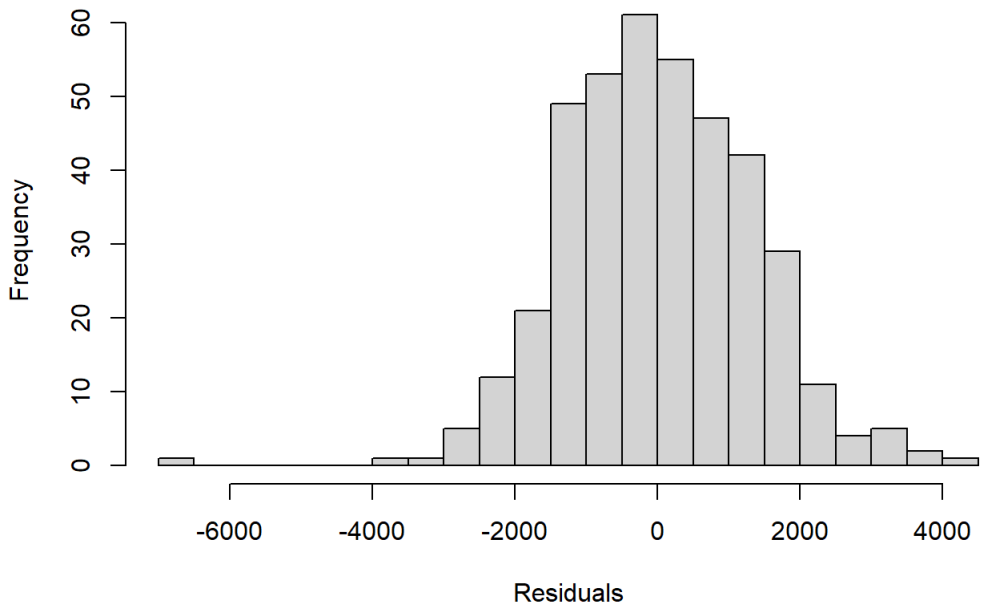
```
car.lm.pred <- predict(car.lm, valid.df)
some.residuals <- valid.df$Price[1:20] - car.lm.pred[1:20]
data.frame("Predicted" = car.lm.pred[1:20], "Actual" = valid.df$Price[1:20],
           "Residual" = some.residuals)
```

```
##    Predicted Actual   Residual
## 2   16447.08  13750 -2697.0840
## 7   16756.56  16900   143.4355
## 8   16749.79  18600  1850.2149
## 9   20959.15  21500   540.8458
## 10  14349.86  12950 -1399.8630
## 12  21123.59  19950 -1173.5906
## 13  20963.53  19600 -1363.5268
## 14  20408.11  21500  1091.8910
## 18  16816.89  17950  1133.1091
## 21  15052.80  15950   897.2004
## 23  15800.08  15950   149.9208
## 24  16306.60  16950   643.4021
## 26  16785.55  15950  -835.5530
## 30  16483.62  17950  1466.3818
## 32  16232.80  15750  -482.7964
## 34  15752.34  14950  -802.3373
## 36  15484.64  15750   265.3647
## 38  16628.51  14950 -1678.5051
## 46  18069.11  19000   930.8855
## 47  17441.14  17950   508.8628
```

```
options(scipen=999, digits = 3)
# use accuracy() to compute common accuracy measures.
accuracy(car.lm.pred, valid.df$Price)
```

```
##            ME RMSE  MAE   MPE MAPE
## Test set 19.6 1325 1049 -0.75 9.35
```

```
car.lm.pred <- predict(car.lm, valid.df)
all.residuals <- valid.df$Price - car.lm.pred
hist(all.residuals, breaks = 25, xlab = "Residuals", main = "")
```



Backward stepwise regression

Backward stepwise regression is a statistical approach employed in regression analysis to construct predictive models systematically by iteratively eliminating independent variables that contribute minimally to explaining the dependent variable's variance. This method begins with a full model encompassing all potential predictor variables and progressively removes the least influential ones. It finds utility in situations with multiple independent variables for several reasons: it simplifies complex models, enhances interpretability, guards against overfitting, and facilitates hypothesis testing by identifying the most significant variables in predicting the outcome of interest. In essence, backward stepwise regression streamlines models while preserving their predictive power, making it a valuable tool in statistical analysis.

```
# use step() to run stepwise regression.
car.lm.step <- step(car.lm, direction = "backward")
```

```
## Start:  AIC=8698
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Automatic +
##     CC + Doors + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - CC             1      23445 1139558987 8696
## - Automatic      1     113837 1139649380 8696
## - Met_Color      1     335154 1139870697 8696
## - Doors          1    1469490 1141005033 8697
## <none>                        1139535543 8698
## - Fuel_Type      2   36864358 1176399900 8713
## - Quarterly_Tax  1   48732676 1188268219 8721
## - HP             1   98229083 1237764626 8746
## - KM             1  160862596 1300398139 8775
## - Weight         1  218676925 1358212468 8802
## - Age_08_04      1 1432472333 2572007876 9185
##
## Step:  AIC=8696
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Automatic +
##     Doors + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Automatic      1     136842 1139695829 8694
## - Met_Color      1     340681 1139899668 8694
## - Doors          1    1457424 1141016411 8695
## <none>                        1139558987 8696
## - Fuel_Type      2   36879383 1176438370 8711
## - Quarterly_Tax  1   48759179 1188318167 8719
## - HP             1  100144734 1239703722 8745
## - KM             1  160839218 1300398206 8773
## - Weight         1  218873160 1358432148 8800
## - Age_08_04      1 1433096756 2572655743 9183
##
## Step:  AIC=8694
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Doors +
##     Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Met_Color      1     338704 1140034533 8692
## - Doors          1    1522740 1141218569 8693
## <none>                        1139695829 8694
## - Fuel_Type      2   37033833 1176729662 8709
## - Quarterly_Tax  1   48735659 1188431487 8717
## - HP             1  100045224 1239741053 8743
## - KM             1  161464457 1301160286 8772
## - Weight         1  226617762 1366313591 8801
## - Age_08_04      1 1440955839 2580651668 9183
##
## Step:  AIC=8692
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Doors + Quarterly_Tax +
##     Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Doors          1    1362886 1141397420 8691
## <none>                        1140034533 8692
## - Fuel_Type      2   36776012 1176810545 8707
## - Quarterly_Tax  1   48499275 1188533808 8715
## - HP             1  101053268 1241087802 8741
## - KM             1  161965108 1301999641 8770
## - Weight         1  226421966 1366456500 8799
## - Age_08_04      1 1448501122 2588535655 9182
##
## Step:  AIC=8691
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Quarterly_Tax + Weight
##
```

```
##                Df  Sum of Sq        RSS  AIC
## <none>                       1141397420 8691
## - Fuel_Type     2   35587030 1176984450 8706
## - Quarterly_Tax 1   48089820 1189487240 8714
## - HP            1  102605929 1244003348 8741
## - KM            1  165583130 1306980550 8770
## - Weight        1  232428680 1373826100 8800
## - Age_08_04     1 1447234462 2588631881 9180
```

```
summary(car.lm.step)  # Which variables were dropped?
```

```
##
## Call:
## lm(formula = Price ~ Age_08_04 + KM + Fuel_Type + HP + Quarterly_Tax +
##     Weight, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9667   -748     21    746   6987
##
## Coefficients:
##                    Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)     -4622.46993  1634.07988   -2.83              0.0048 **
## Age_08_04        -133.13196     4.85926  -27.40 < 0.0000000000000002 ***
## KM                 -0.02120     0.00229   -9.27 < 0.0000000000000002 ***
## Fuel_TypeDiesel   888.54989   596.23572    1.49              0.1367
## Fuel_TypePetrol  2138.33406   571.47519    3.74              0.0002 ***
## HP                 37.60879     5.15538    7.30     0.00000000000096 ***
## Quarterly_Tax      12.97858     2.59871    4.99     0.00000077835339 ***
## Weight             15.96199     1.45378   10.98 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1390 on 592 degrees of freedom
## Multiple R-squared:  0.87,   Adjusted R-squared:  0.869
## F-statistic:  566 on 7 and 592 DF,  p-value: <0.0000000000000002
```

```
car.lm.step.pred <- predict(car.lm.step, valid.df)
accuracy(car.lm.step.pred, valid.df$Price)
```

```
##            ME RMSE  MAE    MPE MAPE
## Test set 20.4 1328 1055 -0.736 9.41
```

Regression model with no predictors

Creating a regression model with no predictors, often called a null model, serves as a baseline for comparison in statistical analysis. It helps assess whether adding predictors significantly improves model performance, aids in hypothesis testing to determine if predictors have a meaningful impact, and simplifies the process of model selection by identifying which variables contribute meaningfully to explaining the dependent variable. Additionally, null models are valuable for control group analysis in experimental settings and for educational purposes to illustrate the concept of model complexity.

```
car.lm.null <- lm(Price~1, data = train.df)
# use step() to run forward regression.
car.lm.step <- step(car.lm.null, scope=list(lower=car.lm.null, upper=car.lm), direction = "forward")
```

```
## Start:  AIC=9902
## Price ~ 1
##
##                 Df  Sum of Sq        RSS  AIC
## + Age_08_04      1 6619336124 2167322714 9064
## + KM             1 3357409684 5429249154 9615
## + Weight         1 3185574106 5601084732 9634
## + HP             1 1030024095 7756634743 9829
## + Quarterly_Tax  1  454459218 8332199620 9872
## + Doors          1  283210488 8503448350 9884
## + Met_Color      1  136926847 8649731991 9894
## + CC             1  132887209 8653771629 9895
## + Fuel_Type      2   82914175 8703744663 9900
## <none>                       8786658838 9902
## + Automatic      1   14099233 8772559605 9903
##
## Step:  AIC=9064
## Price ~ Age_08_04
##
##                 Df Sum of Sq        RSS  AIC
## + Weight         1 367311518 1800011196 8954
## + HP             1 348771443 1818551271 8961
## + KM             1 229319983 1938002731 8999
## + Quarterly_Tax  1  29968151 2137354563 9058
## + Automatic      1  19010241 2148312473 9061
## + Doors          1  17838602 2149484111 9061
## + Fuel_Type      2  24222614 2143100100 9061
## + CC             1  13455747 2153866967 9062
## <none>                       2167322714 9064
## + Met_Color      1   3355998 2163966716 9065
##
## Step:  AIC=8954
## Price ~ Age_08_04 + Weight
##
##                 Df Sum of Sq        RSS  AIC
## + KM             1 428119347 1371891849 8794
## + HP             1 373615357 1426395839 8817
## + Fuel_Type      2 317441967 1482569229 8842
## + Quarterly_Tax  1  66286337 1733724859 8934
## + Automatic      1   8279853 1791731343 8954
## <none>                       1800011196 8954
## + Met_Color      1   2076895 1797934301 8956
## + CC             1    276268 1799734929 8956
## + Doors          1    230044 1799781152 8956
##
## Step:  AIC=8794
## Price ~ Age_08_04 + Weight + KM
##
##                 Df Sum of Sq        RSS  AIC
## + HP             1 170728393 1201163456 8716
## + Fuel_Type      2  65233378 1306658471 8768
## <none>                       1371891849 8794
## + Met_Color      1    718694 1371173155 8795
## + CC             1    551713 1371340136 8795
## + Automatic      1    380438 1371511411 8795
## + Doors          1    119381 1371772468 8795
## + Quarterly_Tax  1     20420 1371871429 8796
##
## Step:  AIC=8716
## Price ~ Age_08_04 + Weight + KM + HP
##
##                 Df Sum of Sq        RSS  AIC
## + Quarterly_Tax  1  24179006 1176984450 8706
## + Fuel_Type      2  11676216 1189487240 8714
## <none>                       1201163456 8716
```

```
## + Doors          1    635682 1200527775 8717
## + CC             1    141287 1201022170 8718
## + Automatic      1     34178 1201129279 8718
## + Met_Color      1     21772 1201141684 8718
##
## Step:  AIC=8706
## Price ~ Age_08_04 + Weight + KM + HP + Quarterly_Tax
##
##             Df Sum of Sq        RSS  AIC
## + Fuel_Type  2  35587030 1141397420 8691
## <none>                   1176984450 8706
## + Automatic  1    314349 1176670101 8707
## + Doors      1    173905 1176810545 8707
## + Met_Color  1     52675 1176931775 8708
## + CC         1     11182 1176973268 8708
##
## Step:  AIC=8691
## Price ~ Age_08_04 + Weight + KM + HP + Quarterly_Tax + Fuel_Type
##
##             Df Sum of Sq        RSS  AIC
## <none>                   1141397420 8691
## + Doors      1   1362886 1140034533 8692
## + Automatic  1    197092 1141200327 8693
## + Met_Color  1    178851 1141218569 8693
## + CC         1     38498 1141358922 8693
```

```
summary(car.lm.step)  # Which variables were added?
```

```
##
## Call:
## lm(formula = Price ~ Age_08_04 + Weight + KM + HP + Quarterly_Tax +
##     Fuel_Type, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9667   -748     21    746   6987
##
## Coefficients:
##                    Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)     -4622.46993  1634.07988   -2.83              0.0048 **
## Age_08_04        -133.13196     4.85926  -27.40 < 0.0000000000000002 ***
## Weight             15.96199     1.45378   10.98 < 0.0000000000000002 ***
## KM                 -0.02120     0.00229   -9.27 < 0.0000000000000002 ***
## HP                 37.60879     5.15538    7.30     0.00000000000096 ***
## Quarterly_Tax      12.97858     2.59871    4.99     0.00000077835339 ***
## Fuel_TypeDiesel   888.54989   596.23572    1.49               0.1367
## Fuel_TypePetrol  2138.33406   571.47519    3.74               0.0002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1390 on 592 degrees of freedom
## Multiple R-squared:  0.87,   Adjusted R-squared:  0.869
## F-statistic:  566 on 7 and 592 DF,  p-value: <0.0000000000000002
```

```
car.lm.step.pred <- predict(car.lm.step, valid.df)
accuracy(car.lm.step.pred, valid.df$Price)
```

```
##            ME RMSE  MAE    MPE MAPE
## Test set 20.4 1328 1055 -0.736 9.41
```

```
# use step() to run stepwise regression.
car.lm.step <- step(car.lm, direction = "both")
```

```
## Start:  AIC=8698
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Automatic +
##     CC + Doors + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - CC             1      23445 1139558987 8696
## - Automatic      1     113837 1139649380 8696
## - Met_Color      1     335154 1139870697 8696
## - Doors          1    1469490 1141005033 8697
## <none>                        1139535543 8698
## - Fuel_Type      2   36864358 1176399900 8713
## - Quarterly_Tax  1   48732676 1188268219 8721
## - HP             1   98229083 1237764626 8746
## - KM             1  160862596 1300398139 8775
## - Weight         1  218676925 1358212468 8802
## - Age_08_04      1 1432472333 2572007876 9185
##
## Step:  AIC=8696
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Automatic +
##     Doors + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Automatic      1     136842 1139695829 8694
## - Met_Color      1     340681 1139899668 8694
## - Doors          1    1457424 1141016411 8695
## <none>                        1139558987 8696
## + CC             1      23445 1139535543 8698
## - Fuel_Type      2   36879383 1176438370 8711
## - Quarterly_Tax  1   48759179 1188318167 8719
## - HP             1  100144734 1239703722 8745
## - KM             1  160839218 1300398206 8773
## - Weight         1  218873160 1358432148 8800
## - Age_08_04      1 1433096756 2572655743 9183
##
## Step:  AIC=8694
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Met_Color + Doors +
##     Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Met_Color      1     338704 1140034533 8692
## - Doors          1    1522740 1141218569 8693
## <none>                        1139695829 8694
## + Automatic      1     136842 1139558987 8696
## + CC             1      46449 1139649380 8696
## - Fuel_Type      2   37033833 1176729662 8709
## - Quarterly_Tax  1   48735659 1188431487 8717
## - HP             1  100045224 1239741053 8743
## - KM             1  161464457 1301160286 8772
## - Weight         1  226617762 1366313591 8801
## - Age_08_04      1 1440955839 2580651668 9183
##
## Step:  AIC=8692
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Doors + Quarterly_Tax +
##     Weight
##
##                 Df  Sum of Sq        RSS  AIC
## - Doors          1    1362886 1141397420 8691
## <none>                        1140034533 8692
## + Met_Color      1     338704 1139695829 8694
## + Automatic      1     134865 1139899668 8694
## + CC             1      53737 1139980796 8694
## - Fuel_Type      2   36776012 1176810545 8707
## - Quarterly_Tax  1   48499275 1188533808 8715
## - HP             1  101053268 1241087802 8741
## - KM             1  161965108 1301999641 8770
```

```
## - Weight         1  226421966 1366456500 8799
## - Age_08_04      1 1448501122 2588535655 9182
##
## Step:  AIC=8691
## Price ~ Age_08_04 + KM + Fuel_Type + HP + Quarterly_Tax + Weight
##
##                 Df  Sum of Sq        RSS  AIC
## <none>                        1141397420 8691
## + Doors          1    1362886 1140034533 8692
## + Automatic      1     197092 1141200327 8693
## + Met_Color      1     178851 1141218569 8693
## + CC             1      38498 1141358922 8693
## - Fuel_Type      2   35587030 1176984450 8706
## - Quarterly_Tax  1   48089820 1189487240 8714
## - HP             1  102605929 1244003348 8741
## - KM             1  165583130 1306980550 8770
## - Weight         1  232428680 1373826100 8800
## - Age_08_04      1 1447234462 2588631881 9180
```

```
summary(car.lm.step)  # Which variables were dropped/added?
```

```
##
## Call:
## lm(formula = Price ~ Age_08_04 + KM + Fuel_Type + HP + Quarterly_Tax +
##     Weight, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9667   -748     21    746   6987
##
## Coefficients:
##                    Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)     -4622.46993  1634.07988   -2.83               0.0048 **
## Age_08_04        -133.13196     4.85926  -27.40 < 0.0000000000000002 ***
## KM                 -0.02120     0.00229   -9.27 < 0.0000000000000002 ***
## Fuel_TypeDiesel   888.54989   596.23572    1.49               0.1367
## Fuel_TypePetrol  2138.33406   571.47519    3.74               0.0002 ***
## HP                 37.60879     5.15538    7.30     0.00000000000096 ***
## Quarterly_Tax      12.97858     2.59871    4.99     0.00000077835339 ***
## Weight             15.96199     1.45378   10.98 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1390 on 592 degrees of freedom
## Multiple R-squared:  0.87,   Adjusted R-squared:  0.869
## F-statistic:  566 on 7 and 592 DF,  p-value: <0.0000000000000002
```

```
car.lm.step.pred <- predict(car.lm.step, valid.df)
accuracy(car.lm.step.pred, valid.df$Price)
```

```
##            ME RMSE  MAE    MPE MAPE
## Test set 20.4 1328 1055 -0.736 9.41
```