Predicting the price of a car using various attribues

Jaymeen Gandhi

February 25, 2020

require(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages ----------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

new\_ads=read\_csv('C:/Users/jayme/Downloads/new\_cars.csv')

## Parsed with column specification:  
## cols(  
## maker = col\_character(),  
## model = col\_character(),  
## mileage = col\_double(),  
## manufacture\_year = col\_double(),  
## engine\_displacement = col\_double(),  
## engine\_power = col\_double(),  
## transmission = col\_character(),  
## door\_count = col\_double(),  
## seat\_count = col\_double(),  
## fuel\_type = col\_character(),  
## date\_created = col\_logical(),  
## date\_last\_seen = col\_logical(),  
## views = col\_logical(),  
## ratings = col\_double(),  
## price\_eur = col\_logical(),  
## id = col\_double()  
## )

classified\_ads= read\_csv('C:/Users/jayme/Downloads/current\_ads.csv')

## Parsed with column specification:  
## cols(  
## maker = col\_character(),  
## model = col\_character(),  
## mileage = col\_double(),  
## manufacture\_year = col\_double(),  
## engine\_displacement = col\_double(),  
## engine\_power = col\_double(),  
## transmission = col\_character(),  
## door\_count = col\_double(),  
## seat\_count = col\_double(),  
## fuel\_type = col\_character(),  
## date\_created = col\_datetime(format = ""),  
## date\_last\_seen = col\_datetime(format = ""),  
## views = col\_double(),  
## ratings = col\_double(),  
## price\_eur = col\_double(),  
## id = col\_double()  
## )

classified\_ads ##(historical data)

## # A tibble: 2,190 x 16  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 207000 1997 1781 92  
## 2 audi a4 62500 2008 1798 118  
## 3 niss~ micra 98500 2003 998 48  
## 4 seat leon 32426 2014 1197 77  
## 5 audi a4 256000 1998 1781 92  
## 6 hyun~ i30 74000 2009 1396 80  
## 7 niss~ alme~ 170000 2005 1769 85  
## 8 skoda supe~ 78130 2011 1968 125  
## 9 opel astra 183200 1996 1598 74  
## 10 audi a3 196500 2006 1968 103  
## # ... with 2,180 more rows, and 10 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <dttm>,  
## # date\_last\_seen <dttm>, views <dbl>, ratings <dbl>, price\_eur <dbl>,  
## # id <dbl>

new\_ads ##(new data)

## # A tibble: 730 x 16  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 228000 2008 1968 100  
## 2 ford focus 247151 2002 1800 55  
## 3 skoda octa~ 14513 2015 1968 135  
## 4 skoda fabia 126880 2005 1390 59  
## 5 skoda octa~ 12591 2014 1958 110  
## 6 ford s-max 177896 2006 1999 107  
## 7 skoda rapid 3650 2015 1390 90  
## 8 citr~ c5 351958 2004 1997 100  
## 9 skoda octa~ 112844 2006 2000 110  
## 10 audi a6 184800 1998 2771 142  
## # ... with 720 more rows, and 10 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <lgl>,  
## # date\_last\_seen <lgl>, views <lgl>, ratings <dbl>, price\_eur <lgl>, id <dbl>

## Get the predicted target (price) in the new data using the historical data.

my\_model=lm(formula = price\_eur ~ mileage , data = classified\_ads)  
summary(my\_model)

##   
## Call:  
## lm(formula = price\_eur ~ mileage, data = classified\_ads)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15986 -6803 -2879 2813 225478   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.008e+04 4.180e+02 48.04 <2e-16 \*\*\*  
## mileage -7.343e-02 3.408e-03 -21.54 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12350 on 2188 degrees of freedom  
## Multiple R-squared: 0.175, Adjusted R-squared: 0.1746   
## F-statistic: 464.1 on 1 and 2188 DF, p-value: < 2.2e-16

f = price\_eur ~ mileage  
#lm (f , data = classified\_ads)

require(broom)

## Loading required package: broom

augment(my\_model , classified\_ads )

## # A tibble: 2,190 x 23  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 207000 1997 1781 92  
## 2 audi a4 62500 2008 1798 118  
## 3 niss~ micra 98500 2003 998 48  
## 4 seat leon 32426 2014 1197 77  
## 5 audi a4 256000 1998 1781 92  
## 6 hyun~ i30 74000 2009 1396 80  
## 7 niss~ alme~ 170000 2005 1769 85  
## 8 skoda supe~ 78130 2011 1968 125  
## 9 opel astra 183200 1996 1598 74  
## 10 audi a3 196500 2006 1968 103  
## # ... with 2,180 more rows, and 17 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <dttm>,  
## # date\_last\_seen <dttm>, views <dbl>, ratings <dbl>, price\_eur <dbl>,  
## # id <dbl>, .fitted <dbl>, .se.fit <dbl>, .resid <dbl>, .hat <dbl>,  
## # .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>

## augment function is just used to calculate the r-square value. It is not used much.  
## If we have the mileage column we can grt the price.

(beta\_zero = my\_model$ coefficients[1])

## (Intercept)   
## 20080.27

(beta\_one = my\_model $ coefficients[2])

## mileage   
## -0.07342887

beta\_zero

## (Intercept)   
## 20080.27

beta\_one

## mileage   
## -0.07342887

## Intecept + coeffeicent of the intercept \* value

## B0 + Mileage \* Value of mileage

20080.27 + -0.07342887 \* 207000

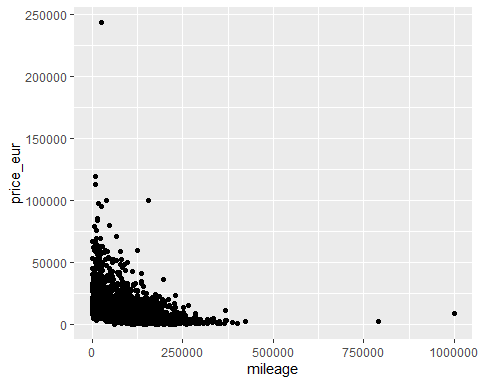
## [1] 4880.494

## Error= Actual Price - Targetted Price

1300.11 - 4880.494

## [1] -3580.384

ggplot(classified\_ads, aes(x = mileage , y= price\_eur)) + geom\_point()



(beta\_zero = my\_model$ coefficients[1])

## (Intercept)   
## 20080.27

(beta\_one = my\_model $ coefficients[2])

## mileage   
## -0.07342887

beta\_zero

## (Intercept)   
## 20080.27

beta\_one

## mileage   
## -0.07342887

## Price = intercept + coeffeicint \* values of mileage

new\_ads %>% mutate(predicted\_price = beta\_zero + beta\_one \* mileage)

## # A tibble: 730 x 17  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 228000 2008 1968 100  
## 2 ford focus 247151 2002 1800 55  
## 3 skoda octa~ 14513 2015 1968 135  
## 4 skoda fabia 126880 2005 1390 59  
## 5 skoda octa~ 12591 2014 1958 110  
## 6 ford s-max 177896 2006 1999 107  
## 7 skoda rapid 3650 2015 1390 90  
## 8 citr~ c5 351958 2004 1997 100  
## 9 skoda octa~ 112844 2006 2000 110  
## 10 audi a6 184800 1998 2771 142  
## # ... with 720 more rows, and 11 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <lgl>,  
## # date\_last\_seen <lgl>, views <lgl>, ratings <dbl>, price\_eur <lgl>,  
## # id <dbl>, predicted\_price <dbl>

predict(my\_model, newdata = new\_ads)

## 1 2 3 4 5 6   
## 3338.48375 1932.24737 19014.59394 10763.61157 19155.72424 7017.56410   
## 7 8 9 10 11 12   
## 19812.25181 -5763.61270 11794.25926 6510.61115 7303.64299 18153.49353   
## 13 14 15 16 17 18   
## 8551.93386 6421.68878 -2682.68398 18569.83525 9065.93599 18134.40202   
## 19 20 21 22 23 24   
## 5122.80541 9433.08036 5100.77675 11369.39979 -39.24448 13596.49756   
## 25 26 27 28 29 30   
## 3191.62601 6894.57073 8327.38836 3632.19925 19492.83620 10826.09954   
## 31 32 33 34 35 36   
## 9212.79374 3948.89799 9487.63801 5467.92112 14820.55690 10828.22898   
## 37 38 39 40 41 42   
## 4513.34575 3172.53450 18168.03245 9494.90747 11156.52948 17422.06851   
## 43 44 45 46 47 48   
## 4236.66575 19530.28493 4195.39872 18377.01102 13854.74691 7424.72721   
## 49 50 51 52 53 54   
## 8737.12149 18494.13008 19353.32134 18773.67381 16504.28100 18260.18568   
## 55 56 57 58 59 60   
## 9755.06598 7630.32806 9910.95548 18879.70510 19676.40839 12223.37760   
## 61 62 63 64 65 66   
## 15307.39034 19125.69183 16335.39459 12745.45690 16908.13981 17510.40344   
## 67 68 69 70 71 72   
## 18453.74420 13618.52622 9378.74299 18009.57293 10093.94023 -112.67336   
## 73 74 75 76 77 78   
## 7510.19842 7905.75977 18035.27304 18430.61410 17187.75697 3705.62813   
## 79 80 81 82 83 84   
## 3485.34150 7303.64299 12260.09204 17436.82771 13655.24066 10959.37295   
## 85 86 87 88 89 90   
## 17637.43540 18552.94661 18714.49013 18373.04586 12781.43705 14826.13750   
## 91 92 93 94 95 96   
## 10656.33199 14147.21412 18934.77676 13398.23960 1135.61751 10095.48224   
## 97 98 99 100 101 102   
## 6022.08884 7670.78737 14793.38822 11309.18811 6874.08408 10754.80011   
## 103 104 105 106 107 108   
## 11412.64940 13471.66847 15357.24855 4660.20350 19999.49544 12406.94979   
## 109 110 111 112 113 114   
## 1282.47526 15438.75460 7817.64512 9979.17090 17950.82983 8551.93386   
## 115 116 117 118 119 120   
## 11268.80223 15843.42113 19015.54852 17475.23101 6039.41806 10378.77084   
## 121 122 123 124 125 126   
## 13102.90866 19151.39194 12540.73720 6685.73901 3975.99325 8570.51137   
## 127 128 129 130 131 132   
## 13741.81330 16930.60905 10718.08567 9800.22473 19915.05223 19258.67152   
## 133 134 135 136 137 138   
## 6349.06762 13545.09735 19970.12389 16555.68121 -5114.72174 6863.06974   
## 139 140 141 142 143 144   
## 17436.82771 10975.08673 10901.65786 11709.37548 19970.12389 7103.91645   
## 145 146 147 148 149 150   
## 17216.54109 8787.71398 9805.36476 17216.54109 12223.37760 18270.24544   
## 151 152 153 154 155 156   
## 18611.68971 18206.28889 6128.78100 18429.58610 13765.38397 19430.42166   
## 157 158 159 160 161 162   
## 17877.40096 8258.21836 14499.67272 12539.12176 15836.07824 13544.95049   
## 163 164 165 166 167 168   
## 18778.37325 19701.08049 7172.13188 12517.09310 16467.71342 14325.86657   
## 169 170 171 172 173 174   
## 6716.21199 13728.66954 15968.25022 19529.55064 19859.98058 9947.08248   
## 175 176 177 178 179 180   
## 18099.15616 12055.00519 -993.81986 6413.24446 9502.69093 19780.38368   
## 181 182 183 184 185 186   
## 19161.67198 19604.30124 5614.77887 9139.36486 19207.27131 18376.71731   
## 187 188 189 190 191 192   
## 19022.89141 12803.24543 2530.76613 11401.56164 14007.25869 4329.77356   
## 193 194 195 196 197 198   
## 14279.38610 19353.32134 10850.25764 15904.07338 19382.69289 9095.30754   
## 199 200 201 202 203 204   
## 3265.05488 14132.52835 7964.50287 19342.30701 4284.24766 9580.23183   
## 205 206 207 208 209 210   
## -1625.38161 14113.43684 17519.06805 1209.04639 19492.83620 18612.42400   
## 211 212 213 214 215 216   
## 11635.94660 14088.47102 11415.65998 10647.52052 12076.51985 9873.65361   
## 217 218 219 220 221 222   
## 8059.96040 6752.92643 6936.49862 3632.19925 19621.41016 12168.30595   
## 223 224 225 226 227 228   
## 18427.01609 16269.30860 18281.25977 18699.80436 19786.55170 14631.33069   
## 229 230 231 232 233 234   
## 17912.94054 19808.58037 13765.38397 19933.40945 6936.49862 12937.17969   
## 235 236 237 238 239 240   
## 12377.57824 7028.28471 15949.81957 9212.79374 3364.91815 19463.46465   
## 241 242 243 244 245 246   
## 12491.39300 19147.42678 10101.28312 18244.54533 12387.27085 15013.67484   
## 247 248 249 250 251 252   
## 3290.38784 15028.36062 15601.10584 13691.95510 16225.25128 13827.79852   
## 253 254 255 256 257 258   
## 14925.56019 5739.60796 10901.65786 7303.64299 7237.55701 7817.64512   
## 259 260 261 262 263 264   
## 18184.70080 14998.98907 17912.20625 11239.43068 19199.12071 16115.10797   
## 265 266 267 268 269 270   
## 17561.73023 16078.39353 -4346.21513 8185.52378 14205.95722 18472.83571   
## 271 272 273 274 275 276   
## 15307.39034 17282.62707 19960.35785 9873.65361 16721.92419 -1214.10648   
## 277 278 279 280 281 282   
## 13104.52410 14205.95722 -479.81773 16746.59629 13691.95510 10372.96996   
## 283 284 285 286 287 288   
## 15296.37601 7707.42837 13798.05982 15233.96147 11031.03953 -1140.67761   
## 289 290 291 292 293 294   
## 12355.32929 13178.46698 19412.06444 12708.00818 11770.32144 18501.54640   
## 295 296 297 298 299 300   
## 10314.22686 18520.93162 19088.97739 12003.09098 10908.26646 15065.07506   
## 301 302 303 304 305 306   
## 18985.00211 18414.90032 17583.68546 12076.51985 15446.90520 18126.98571   
## 307 308 309 310 311 312   
## 16416.75378 16614.71803 12517.09310 9286.22261 15481.34335 946.17101   
## 313 314 315 316 317 318   
## 14220.64300 9506.50924 7964.50287 15490.96253 16454.93680 10462.55319   
## 319 320 321 322 323 324   
## 10020.51136 2090.19288 7971.84575 -553.24661 9506.50924 16922.82559   
## 325 326 327 328 329 330   
## 12737.37973 4219.63025 17958.17272 3462.65198 8184.78949 9682.59168   
## 331 332 333 334 335 336   
## 11268.80223 12223.37760 16156.59528 7377.07187 9506.50924 9433.08036   
## 337 338 339 340 341 342   
## 12480.37866 19398.92067 14205.95722 5962.02402 10681.37123 2751.05276   
## 343 344 345 346 347 348   
## 7083.35637 12260.09204 16911.81126 9183.42219 10538.18493 12003.09098   
## 349 350 351 352 353 354   
## 12722.69395 18097.68758 19000.86274 9286.22261 10754.80011 10167.36911   
## 355 356 357 358 359 360   
## 17510.25659 7670.78737 15200.47790 11085.23004 3632.19925 12059.55778   
## 361 362 363 364 365 366   
## 19544.23642 18053.63026 18399.48026 19269.24528 15432.21943 6716.21199   
## 367 368 369 370 371 372   
## 7597.35849 10167.36911 12003.09098 1649.61964 16739.69397 11621.26083   
## 373 374 375 376 377 378   
## 9800.22473 12296.80648 17914.11540 -692.39433 -10833.28908 19353.32134   
## 379 380 381 382 383 384   
## 15444.84920 13471.66847 19133.03472 18097.68758 13467.18931 10850.25764   
## 385 386 387 388 389 390   
## 6339.08129 19487.98990 9565.25234 17311.99862 18850.62727 19819.88841   
## 391 392 393 394 395 396   
## 10461.08461 18273.91688 13324.81072 17560.33508 16996.25446 12370.23535   
## 397 398 399 400 401 402   
## 15314.73323 19489.23819 9029.22155 6202.20987 12370.23535 12821.82293   
## 403 404 405 406 407 408   
## 18324.87652 12575.83620 14088.47102 18244.69219 17032.96890 15681.87760   
## 409 410 411 412 413 414   
## 7083.35637 2310.47951 5981.92325 -1948.39523 10938.37229 17243.34263   
## 415 416 417 418 419 420   
## 18827.05660 19786.55170 18545.60372 16070.97721 11771.79002 14279.38610   
## 421 422 423 424 425 426   
## 7428.47208 15960.68704 17730.54321 19345.97846 16886.11115 19022.89141   
## 427 428 429 430 431 432   
## 10990.43337 4072.77250 11672.66104 12202.59723 16188.53684 18609.85399   
## 433 434 435 436 437 438   
## 15503.07829 7597.35849 17877.40096 7303.64299 13673.23073 14242.67166   
## 439 440 441 442 443 444   
## 12223.37760 19298.24969 5467.92112 11694.68970 7230.21412 19845.58852   
## 445 446 447 448 449 450   
## 16041.67909 14756.67378 18112.37336 7597.35849 12994.38079 12780.26219   
## 451 452 453 454 455 456   
## 16335.39459 18213.85206 7428.47208 16864.08249 14719.95934 14646.53047   
## 457 458 459 460 461 462   
## 17715.85744 15448.37378 15067.05764 16041.67909 7845.40123 16335.39459   
## 463 464 465 466 467 468   
## 14374.84363 6055.35212 16423.50924 107.61326 12660.42626 7744.21624   
## 469 470 471 472 473 474   
## 18662.28220 1723.12194 12662.99628 10255.48376 107.61326 9065.93599   
## 475 476 477 478 479 480   
## 15527.67697 4660.20350 9800.22473 5527.32508 14719.95934 12456.80800   
## 481 482 483 484 485 486   
## 10770.95446 13141.23854 10902.68586 18155.76982 13169.50865 19500.17909   
## 487 488 489 490 491 492   
## 9359.65149 16041.67909 6412.65703 16266.22459 16755.04061 16397.80913   
## 493 494 495 496 497 498   
## 9818.58195 8236.63028 11669.72389 16115.10797 2677.62388 8625.36274   
## 499 500 501 502 503 504   
## 8059.96040 19262.26954 6275.63875 18858.70444 18024.25871 18846.66211   
## 505 506 507 508 509 510   
## 17751.91101 19632.35107 3925.91475 14208.16009 6789.64087 8845.64936   
## 511 512 513 514 515 516   
## 11782.80435 4683.03988 10314.22686 17937.09863 13691.95510 12830.04697   
## 517 518 519 520 521 522   
## 7846.42923 7722.26101 7150.17664 12803.39228 13031.09522 19147.20649   
## 523 524 525 526 527 528   
## 12296.80648 5262.09999 8331.64724 18541.05113 19355.37735 8473.07125   
## 529 530 531 532 533 534   
## 18996.23672 17980.20138 15859.94263 18244.54533 18530.91795 15527.67697   
## 535 536 537 538 539 540   
## 19933.40945 19760.55788 4146.20138 18244.54533 3737.12912 16230.53816   
## 541 542 543 544 545 546   
## 14619.65550 13949.39673 9690.08142 10828.22898 16603.40998 17626.64135   
## 547 548 549 550 551 552   
## 18028.29730 17807.64353 12957.66635 8625.36274 10681.37123 15233.96147   
## 553 554 555 556 557 558   
## 10424.37017 19111.00606 19837.95192 10901.65786 18917.52097 16570.36699   
## 559 560 561 562 563 564   
## 11497.23946 18614.92058 16283.18666 18890.71943 13948.95616 -1063.65072   
## 565 566 567 568 569 570   
## 7964.50287 9433.08036 19713.12283 5688.20775 9065.93599 -5472.98122   
## 571 572 573 574 575 576   
## 4880.49013 13985.67060 5541.35000 13618.52622 18446.47474 16880.53056   
## 577 578 579 580 581 582   
## 19524.19033 7156.78524 19904.03790 16694.82893 10895.71012 11342.23111   
## 583 584 585 586 587 588   
## 14599.31570 16078.39353 8845.64936 17817.04242 8595.99119 8331.64724   
## 589 590 591 592 593 594   
## 13796.07724 8845.64936 9653.36699 19859.98058 14573.10160 15415.69793   
## 595 596 597 598 599 600   
## 5394.49225 18097.68758 19529.55064 18182.13079 7009.92749 11195.37336   
## 601 602 603 604 605 606   
## 19275.26645 10640.25106 9074.01316 7494.55807 14995.31762 19346.93303   
## 607 608 609 610 611 612   
## 17220.21253 18846.73554 9076.06917 14478.89235 10607.94236 13012.00372   
## 613 614 615 616 617 618   
## 18942.11964 10607.94236 18365.62955 5600.09310 16988.91158 17263.75585   
## 619 620 621 622 623 624   
## 14996.78620 6598.72579 18372.75215 7830.05460 19345.97846 15178.88981   
## 625 626 627 628 629 630   
## 7913.76351 12737.37973 13530.41157 13398.23960 16937.51136 12296.80648   
## 631 632 633 634 635 636   
## 7744.21624 18251.88822 11833.83742 16444.06932 17730.54321 15233.96147   
## 637 638 639 640 641 642   
## 17855.37230 18997.92559 15164.20404 16445.53790 18538.26083 9972.41545   
## 643 644 645 646 647 648   
## 5148.87266 18156.43068 19529.55064 17253.25552 19309.26402 5614.77887   
## 649 650 651 652 653 654   
## 14539.69146 18373.78015 9800.22473 564.63458 -259.53111 10101.79712   
## 655 656 657 658 659 660   
## 4160.88715 -2535.82623 18493.10207 14059.09947 18811.04910 18578.35300   
## 661 662 663 664 665 666   
## 16785.73388 18053.63026 16247.27994 12774.38788 18721.83302 12066.75381   
## 667 668 669 670 671 672   
## 11540.48907 -839.10522 9873.65361 17987.54427 15281.39652 17693.82877   
## 673 674 675 676 677 678   
## 16696.95837 19162.40627 4787.30888 17863.74319 11694.68970 15196.14560   
## 679 680 681 682 683 684   
## 17244.81120 18731.89278 7450.50074 7817.64512 2111.34040 19153.59480   
## 685 686 687 688 689 690   
## 10534.51348 14925.56019 17298.41428 17304.65574 13618.52622 4439.91688   
## 691 692 693 694 695 696   
## 4293.05913 18681.52057 18790.70930 19529.55064 4660.20350 3191.62601   
## 697 698 699 700 701 702   
## 7964.50287 14826.94522 5405.50658 4733.63238 6569.35424 19071.06075   
## 703 704 705 706 707 708   
## 15535.75414 15835.78453 8919.07824 9800.22473 6863.06974 19536.89353   
## 709 710 711 712 713 714   
## 9065.93599 19015.54852 6569.35424 17877.40096 17857.42831 6973.21306   
## 715 716 717 718 719 720   
## 13031.09522 7230.21412 6260.95297 -3416.97272 16445.53790 16555.68121   
## 721 722 723 724 725 726   
## 14749.33089 9653.36699 18865.60676 19883.11067 17686.48589 17151.26282   
## 727 728 729 730   
## 18091.88670 12549.25495 9065.93599 19434.09310

##### New Model

## Creating a model

my\_model2=lm(formula = price\_eur ~ ratings , data = classified\_ads)  
summary(my\_model2)

##   
## Call:  
## lm(formula = price\_eur ~ ratings, data = classified\_ads)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13188 -8413 -3353 3554 230235   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12697.86 546.33 23.242 <2e-16 \*\*\*  
## ratings 78.99 91.62 0.862 0.389   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 13590 on 2188 degrees of freedom  
## Multiple R-squared: 0.0003396, Adjusted R-squared: -0.0001173   
## F-statistic: 0.7433 on 1 and 2188 DF, p-value: 0.3887

## Got the intercept and coefficint

beta\_zero = my\_model2$ coefficients[1]  
beta\_one = my\_model2$ coefficients[2]  
beta\_zero

## (Intercept)   
## 12697.86

beta\_one

## ratings   
## 78.99483

## Substitute it to the formula

new\_ads %>% mutate(predicted\_price = beta\_zero + beta\_one \* ratings)

## # A tibble: 730 x 17  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 228000 2008 1968 100  
## 2 ford focus 247151 2002 1800 55  
## 3 skoda octa~ 14513 2015 1968 135  
## 4 skoda fabia 126880 2005 1390 59  
## 5 skoda octa~ 12591 2014 1958 110  
## 6 ford s-max 177896 2006 1999 107  
## 7 skoda rapid 3650 2015 1390 90  
## 8 citr~ c5 351958 2004 1997 100  
## 9 skoda octa~ 112844 2006 2000 110  
## 10 audi a6 184800 1998 2771 142  
## # ... with 720 more rows, and 11 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <lgl>,  
## # date\_last\_seen <lgl>, views <lgl>, ratings <dbl>, price\_eur <lgl>,  
## # id <dbl>, predicted\_price <dbl>

## Get the values

predict(my\_model2, newdata = new\_ads)

## 1 2 3 4 5 6 7 8   
## 12855.85 13250.82 13329.82 13329.82 13171.83 12934.85 12934.85 13013.84   
## 9 10 11 12 13 14 15 16   
## 13171.83 12855.85 12934.85 12776.86 13250.82 12776.86 13408.81 13329.82   
## 17 18 19 20 21 22 23 24   
## 13250.82 13408.81 12776.86 13092.84 12697.86 13013.84 13250.82 13329.82   
## 25 26 27 28 29 30 31 32   
## 13250.82 12697.86 13250.82 13250.82 13092.84 12934.85 13013.84 12776.86   
## 33 34 35 36 37 38 39 40   
## 12776.86 13487.81 13329.82 12776.86 13408.81 13329.82 13171.83 13250.82   
## 41 42 43 44 45 46 47 48   
## 13092.84 12855.85 13329.82 13487.81 13250.82 13408.81 12934.85 13092.84   
## 49 50 51 52 53 54 55 56   
## 12855.85 12855.85 12697.86 12934.85 13329.82 12776.86 13171.83 13092.84   
## 57 58 59 60 61 62 63 64   
## 13171.83 12934.85 13092.84 12855.85 12776.86 13171.83 13408.81 12776.86   
## 65 66 67 68 69 70 71 72   
## 13329.82 13250.82 13250.82 12697.86 13250.82 13487.81 13250.82 12776.86   
## 73 74 75 76 77 78 79 80   
## 12697.86 12855.85 13329.82 13171.83 13329.82 13329.82 13329.82 13171.83   
## 81 82 83 84 85 86 87 88   
## 13487.81 13171.83 13250.82 13013.84 12697.86 13171.83 12697.86 13013.84   
## 89 90 91 92 93 94 95 96   
## 13408.81 12776.86 12776.86 13013.84 12697.86 13487.81 13250.82 13171.83   
## 97 98 99 100 101 102 103 104   
## 13092.84 13250.82 13092.84 13408.81 13013.84 12855.85 12776.86 13408.81   
## 105 106 107 108 109 110 111 112   
## 13250.82 13013.84 13171.83 12776.86 13408.81 13408.81 13487.81 13408.81   
## 113 114 115 116 117 118 119 120   
## 12855.85 13171.83 13487.81 13013.84 13250.82 12934.85 13408.81 13013.84   
## 121 122 123 124 125 126 127 128   
## 12776.86 13408.81 13329.82 13487.81 13013.84 12855.85 12776.86 13171.83   
## 129 130 131 132 133 134 135 136   
## 13092.84 13171.83 13013.84 12697.86 13408.81 12855.85 13092.84 13092.84   
## 137 138 139 140 141 142 143 144   
## 13250.82 12855.85 13171.83 12697.86 13408.81 12697.86 13408.81 13408.81   
## 145 146 147 148 149 150 151 152   
## 12776.86 12697.86 13092.84 13250.82 13487.81 13092.84 13487.81 13013.84   
## 153 154 155 156 157 158 159 160   
## 12934.85 12855.85 13487.81 13408.81 13487.81 12697.86 13013.84 12697.86   
## 161 162 163 164 165 166 167 168   
## 13487.81 13329.82 12697.86 13013.84 12855.85 13092.84 13013.84 13171.83   
## 169 170 171 172 173 174 175 176   
## 13329.82 12776.86 12776.86 13250.82 13013.84 12697.86 13408.81 13250.82   
## 177 178 179 180 181 182 183 184   
## 13013.84 13408.81 13171.83 13092.84 12855.85 13092.84 12934.85 13171.83   
## 185 186 187 188 189 190 191 192   
## 13408.81 12697.86 13092.84 13329.82 12776.86 12855.85 13013.84 13250.82   
## 193 194 195 196 197 198 199 200   
## 13329.82 13013.84 12855.85 12697.86 13013.84 12776.86 12855.85 12697.86   
## 201 202 203 204 205 206 207 208   
## 13013.84 13408.81 12776.86 13171.83 13487.81 13171.83 13487.81 13092.84   
## 209 210 211 212 213 214 215 216   
## 13092.84 13171.83 12776.86 12776.86 13487.81 12776.86 13329.82 13329.82   
## 217 218 219 220 221 222 223 224   
## 12855.85 12697.86 13329.82 13013.84 13329.82 13092.84 13092.84 12776.86   
## 225 226 227 228 229 230 231 232   
## 12776.86 13250.82 12697.86 13171.83 13171.83 12697.86 12697.86 13329.82   
## 233 234 235 236 237 238 239 240   
## 13013.84 13092.84 12776.86 12855.85 12697.86 13171.83 13487.81 12776.86   
## 241 242 243 244 245 246 247 248   
## 13250.82 13171.83 12697.86 12855.85 12855.85 12697.86 13250.82 13013.84   
## 249 250 251 252 253 254 255 256   
## 12855.85 13171.83 12855.85 13487.81 12855.85 12855.85 13013.84 13487.81   
## 257 258 259 260 261 262 263 264   
## 13171.83 13487.81 13250.82 12697.86 13250.82 13092.84 13408.81 12697.86   
## 265 266 267 268 269 270 271 272   
## 13487.81 12776.86 13171.83 13171.83 13408.81 13487.81 12697.86 13250.82   
## 273 274 275 276 277 278 279 280   
## 13487.81 13092.84 12776.86 12855.85 13092.84 13171.83 13408.81 12697.86   
## 281 282 283 284 285 286 287 288   
## 12934.85 12934.85 13329.82 12855.85 13092.84 13408.81 13092.84 13487.81   
## 289 290 291 292 293 294 295 296   
## 12934.85 12855.85 13171.83 13329.82 13408.81 13013.84 12934.85 13329.82   
## 297 298 299 300 301 302 303 304   
## 12855.85 13013.84 13013.84 13013.84 13171.83 13408.81 13171.83 13092.84   
## 305 306 307 308 309 310 311 312   
## 13487.81 13092.84 13013.84 13250.82 13329.82 13013.84 13250.82 13171.83   
## 313 314 315 316 317 318 319 320   
## 12855.85 13171.83 12934.85 13408.81 12697.86 13013.84 13329.82 13171.83   
## 321 322 323 324 325 326 327 328   
## 13329.82 13408.81 13013.84 12855.85 12697.86 12855.85 13487.81 13171.83   
## 329 330 331 332 333 334 335 336   
## 13250.82 12855.85 13013.84 12855.85 12855.85 12855.85 13092.84 12934.85   
## 337 338 339 340 341 342 343 344   
## 13092.84 13329.82 13092.84 12855.85 13329.82 12697.86 13487.81 12776.86   
## 345 346 347 348 349 350 351 352   
## 12776.86 12934.85 13487.81 13013.84 13013.84 13250.82 13487.81 13171.83   
## 353 354 355 356 357 358 359 360   
## 13487.81 13329.82 13487.81 12934.85 13487.81 13487.81 12855.85 13408.81   
## 361 362 363 364 365 366 367 368   
## 12697.86 12697.86 13487.81 12855.85 12697.86 12934.85 12855.85 12934.85   
## 369 370 371 372 373 374 375 376   
## 13250.82 13171.83 12934.85 13487.81 13171.83 13171.83 12855.85 13171.83   
## 377 378 379 380 381 382 383 384   
## 12697.86 13171.83 13092.84 12776.86 13171.83 13092.84 13171.83 13408.81   
## 385 386 387 388 389 390 391 392   
## 12855.85 13013.84 12776.86 13408.81 13250.82 12934.85 12934.85 13171.83   
## 393 394 395 396 397 398 399 400   
## 13013.84 13092.84 13487.81 13171.83 13408.81 13408.81 12855.85 12697.86   
## 401 402 403 404 405 406 407 408   
## 12934.85 13013.84 13171.83 13092.84 13408.81 13013.84 13013.84 13487.81   
## 409 410 411 412 413 414 415 416   
## 13250.82 12934.85 13250.82 12855.85 12776.86 12855.85 13171.83 13487.81   
## 417 418 419 420 421 422 423 424   
## 13329.82 13408.81 13013.84 13250.82 13329.82 13487.81 13487.81 12855.85   
## 425 426 427 428 429 430 431 432   
## 13487.81 12934.85 12855.85 12697.86 12934.85 13408.81 13487.81 12776.86   
## 433 434 435 436 437 438 439 440   
## 12934.85 13092.84 12934.85 12697.86 12697.86 13408.81 13013.84 12776.86   
## 441 442 443 444 445 446 447 448   
## 12934.85 13329.82 12934.85 12855.85 12776.86 13408.81 13092.84 13092.84   
## 449 450 451 452 453 454 455 456   
## 13013.84 13487.81 12934.85 12934.85 13408.81 13171.83 13171.83 12934.85   
## 457 458 459 460 461 462 463 464   
## 12855.85 13329.82 13250.82 12776.86 13408.81 12855.85 13171.83 13092.84   
## 465 466 467 468 469 470 471 472   
## 12776.86 12855.85 13329.82 13487.81 13250.82 12934.85 13250.82 12776.86   
## 473 474 475 476 477 478 479 480   
## 12934.85 12855.85 13092.84 13013.84 12934.85 12934.85 12697.86 12934.85   
## 481 482 483 484 485 486 487 488   
## 12855.85 13171.83 12855.85 13408.81 13092.84 13013.84 13013.84 13408.81   
## 489 490 491 492 493 494 495 496   
## 12697.86 13171.83 13171.83 13171.83 13329.82 13250.82 13329.82 12776.86   
## 497 498 499 500 501 502 503 504   
## 13250.82 12855.85 13013.84 12776.86 12776.86 12934.85 12855.85 13013.84   
## 505 506 507 508 509 510 511 512   
## 12776.86 13250.82 12697.86 12697.86 12855.85 13250.82 13250.82 12855.85   
## 513 514 515 516 517 518 519 520   
## 12697.86 13408.81 13487.81 12855.85 13487.81 13092.84 13329.82 12855.85   
## 521 522 523 524 525 526 527 528   
## 13329.82 12776.86 13250.82 12855.85 13013.84 12697.86 13250.82 12855.85   
## 529 530 531 532 533 534 535 536   
## 12697.86 13487.81 12697.86 12776.86 13250.82 13092.84 13171.83 12776.86   
## 537 538 539 540 541 542 543 544   
## 12697.86 13487.81 13250.82 12855.85 12934.85 12855.85 13092.84 13092.84   
## 545 546 547 548 549 550 551 552   
## 12934.85 13092.84 13329.82 13329.82 13171.83 13487.81 13250.82 12697.86   
## 553 554 555 556 557 558 559 560   
## 13329.82 12697.86 13171.83 13092.84 13171.83 13487.81 13250.82 13013.84   
## 561 562 563 564 565 566 567 568   
## 13408.81 13092.84 13408.81 13487.81 12697.86 13329.82 13013.84 12934.85   
## 569 570 571 572 573 574 575 576   
## 13408.81 13487.81 13013.84 12934.85 13329.82 13013.84 13487.81 13013.84   
## 577 578 579 580 581 582 583 584   
## 13487.81 12934.85 13092.84 13250.82 13092.84 13092.84 12697.86 12776.86   
## 585 586 587 588 589 590 591 592   
## 12934.85 12776.86 13092.84 13408.81 12855.85 13013.84 12855.85 13329.82   
## 593 594 595 596 597 598 599 600   
## 13329.82 13250.82 13250.82 13250.82 13171.83 13329.82 13329.82 12697.86   
## 601 602 603 604 605 606 607 608   
## 12776.86 13013.84 13329.82 12934.85 13250.82 13329.82 13329.82 12697.86   
## 609 610 611 612 613 614 615 616   
## 13408.81 13408.81 13487.81 13171.83 13487.81 12776.86 13092.84 12697.86   
## 617 618 619 620 621 622 623 624   
## 13171.83 12697.86 13408.81 13171.83 12697.86 12855.85 13250.82 13487.81   
## 625 626 627 628 629 630 631 632   
## 13487.81 12697.86 13487.81 13408.81 12776.86 12934.85 12776.86 13250.82   
## 633 634 635 636 637 638 639 640   
## 13171.83 13408.81 12776.86 13250.82 12934.85 13329.82 12855.85 13013.84   
## 641 642 643 644 645 646 647 648   
## 13250.82 13092.84 13171.83 12697.86 12776.86 12776.86 12697.86 12697.86   
## 649 650 651 652 653 654 655 656   
## 13013.84 13092.84 13171.83 13171.83 13092.84 12697.86 12697.86 12934.85   
## 657 658 659 660 661 662 663 664   
## 13487.81 13487.81 13092.84 13250.82 12934.85 13250.82 12697.86 13171.83   
## 665 666 667 668 669 670 671 672   
## 12855.85 12855.85 13408.81 13408.81 13013.84 12934.85 12855.85 13013.84   
## 673 674 675 676 677 678 679 680   
## 12934.85 12855.85 13013.84 13092.84 12776.86 12855.85 12697.86 12934.85   
## 681 682 683 684 685 686 687 688   
## 13092.84 12697.86 12855.85 13171.83 12934.85 13408.81 13092.84 13408.81   
## 689 690 691 692 693 694 695 696   
## 13487.81 13250.82 12697.86 13171.83 13013.84 13092.84 12697.86 12855.85   
## 697 698 699 700 701 702 703 704   
## 12697.86 12934.85 13171.83 12776.86 13013.84 13092.84 13092.84 12697.86   
## 705 706 707 708 709 710 711 712   
## 12855.85 13408.81 12697.86 13013.84 13013.84 12697.86 13250.82 12776.86   
## 713 714 715 716 717 718 719 720   
## 13171.83 13487.81 13092.84 13171.83 13408.81 12697.86 13408.81 13408.81   
## 721 722 723 724 725 726 727 728   
## 13408.81 13171.83 12855.85 13329.82 13171.83 13408.81 12776.86 12855.85   
## 729 730   
## 12855.85 13329.82

classified\_ads %>% mutate(fitted\_values= my\_model$fitted.values)

## # A tibble: 2,190 x 17  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 207000 1997 1781 92  
## 2 audi a4 62500 2008 1798 118  
## 3 niss~ micra 98500 2003 998 48  
## 4 seat leon 32426 2014 1197 77  
## 5 audi a4 256000 1998 1781 92  
## 6 hyun~ i30 74000 2009 1396 80  
## 7 niss~ alme~ 170000 2005 1769 85  
## 8 skoda supe~ 78130 2011 1968 125  
## 9 opel astra 183200 1996 1598 74  
## 10 audi a3 196500 2006 1968 103  
## # ... with 2,180 more rows, and 11 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <dttm>,  
## # date\_last\_seen <dttm>, views <dbl>, ratings <dbl>, price\_eur <dbl>,  
## # id <dbl>, fitted\_values <dbl>

require(modelr)

## Loading required package: modelr

##   
## Attaching package: 'modelr'

## The following object is masked from 'package:broom':  
##   
## bootstrap

rsquare(my\_model,classified\_ads)

## [1] 0.1749997

##Another way (Best way)

set.seed(2333)  
training\_data=classified\_ads %>% sample\_frac(size = .7)  
testing\_data = anti\_join(classified\_ads,training\_data,by='id')

f= price\_eur ~ mileage  
model3 = lm(f, training\_data)  
summary(model3)

##   
## Call:  
## lm(formula = f, data = training\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15037 -6917 -2898 2822 225781   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.966e+04 5.021e+02 39.16 <2e-16 \*\*\*  
## mileage -6.889e-02 4.085e-03 -16.86 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12620 on 1531 degrees of freedom  
## Multiple R-squared: 0.1567, Adjusted R-squared: 0.1561   
## F-statistic: 284.4 on 1 and 1531 DF, p-value: < 2.2e-16

rsquare(model3,training\_data)

## [1] 0.1566716

testing\_prediction = predict(model3 , testing\_data)  
testing\_data

## # A tibble: 657 x 16  
## maker model mileage manufacture\_year engine\_displace~ engine\_power  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 audi a4 207000 1997 1781 92  
## 2 niss~ micra 98500 2003 998 48  
## 3 hyun~ i30 74000 2009 1396 80  
## 4 skoda supe~ 78130 2011 1968 125  
## 5 audi a3 75800 2010 1197 77  
## 6 opel astra 5000 2015 999 77  
## 7 audi tt 280871 2000 1781 132  
## 8 pors~ caye~ 136600 2010 4806 368  
## 9 seat ibiza 129283 2009 1390 63  
## 10 skoda octa~ 124500 2012 1968 103  
## # ... with 647 more rows, and 10 more variables: transmission <chr>,  
## # door\_count <dbl>, seat\_count <dbl>, fuel\_type <chr>, date\_created <dttm>,  
## # date\_last\_seen <dttm>, views <dbl>, ratings <dbl>, price\_eur <dbl>,  
## # id <dbl>