CarDekho: A regression approach to vehicle selling price prediction

Written by: Chi-En Chang, Shuang Gao, Gelareh Dejdar

Abstract:
This study intends to investigate the relationships between multiple features of a used car and its selling price using different linear regression models. Branding and other exogenous factors such as income, and demographics are ignored in our case. It was found that the quadratic model produces the best result with RMSE = 2340.42 and $R^2 = 0.919$. We hope to use this result to provide consumers with safer and better understandings of a second hand car when potentially purchasing a used car.

Introduction

CarDekho is an Indian based automobile dealer that provides multiple services online and in person. CarDekho focuses on providing the best deal on a range of different cars based on different factors such as year of car, model, mileage etc. As there is a huge demand for used cars in the Indian Market, CarDekho has continued to grow in these past few years. As of 2020, CarDekho works actively with over 4000 new auto dealerships and 3000 used dealerships across India. The decrease in new car sales seems to be directly related to the increased demand in pre-owned cars - it has reached the point where most car sellers replace their cars with pre-owned rather than purchasing a new car. New cars are mostly priced on original equipment manufacturers but that is not usually the case with used cars, making the pricing of used cars much more complex. Due to this, pricing schemes on used cars is a crucial process and dependent on a numerous set of factors. Our goal is to understand what factors influence the price of used cars in order to create a model which can accurately estimate the price.

Similar studies have been conducted around the world across different fields in the industry. For example, a study published in the Journal of Applied Econometrics, investigated the relationships between miles driven of used cars and their selling prices. The research team first applied an OLS model to investigate the linear relationship of used car prices and annual miles driven. Then, fitted a non-linear structural model to investigate the nonlinear relationships between car price and annual miles driven[1]. They found that the structural model performed better in explaining the variation of the observed response (car prices)[1]. Another study focused on the prediction of used car prices using solely regression models to investigate the relationships of multiple features (engine power, wheels, seats etc.) of the car and their deal prices[2]. They discovered that tree based regression models performed best when predicting used car prices[2].

Our study focuses on price predictions using different linear regression models to identify the best model that fits our data. With this data, we hope to help dealers and consumers find the most reasonable price, depending on their demands in their purchase of secondhand vehicles.

[1] Engers, M., Hartmann, M., & Stern, S. (2009). Annual miles drive used car prices. *Journal of Applied Econometrics*, *24*(1), 1-33.

[2] N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), 2018, pp. 115-119, doi: 10.1109/ICBIR.2018.8391177.

Data Description

The data used in this report was acquired through kaggle.com, a Data Science website that has over 50,000 datasets. The dataset we used includes over 8000 rows of data with 13 features. The first feature is name, a categorical column filled with the names to each corresponding car sold at this dealership. The year column corresponds to the year which the car was bought, ranging from 1983 to 2020. “Selling price” is the price the owner wants to sell the car at, this originally was in indian rupees but we converted it to CAD. The column “Km driven” is the distance completed by the car in km. “Fuel” relates to the type of fuel the car takes, this is a categorical variable. “Seller” type is the categorical feature that defines whether the sale was completed by a dealer or an individual. The “Transmission” feature is another categorical data that indicates whether the car is manual or automatic. The “Owner” feature defines the number of owners the car has previously had. “Mileage” which indicates the car’s fuel efficiency measured in kilometers per liter of gas, and **“**Engine" which is the engine capacity of the car. The “Eax power” and torque columns are two measurements of the engine powers. The last feature is “Eeats” which contain values of how many seats the vehicle has.

Data Preprocessing

We are interested in finding the relationship between price and the respective characteristics that affect the performance of the car (i.e. the make, features, year etc.) hence we first removed the column containing the brand of the cars. As there were very few null values, we decided to remove all the rows that contained null values as it would not affect our dataset too much and would allow for a complete set. The dataset also had information on the car’s engine, mileage driven, max power output and torque. These columns contained both the specific values and the associated units. We are only interested in the numerical values so we removed the unit characters by looping a regular expression pattern to isolate the digits. We then converted the selling price from Indian rupees to CAD with a fixed exchange rate (1 rupee = 0.017 CAD). We also changed the categorical variables to factors.

Exploratory Data Analysis

After data processing, we dive deeper into the dataset to gain a better understanding of the attributes we are modeling on. First we start by examining the summary statistics of the data.

	year	selling_price	km_driven	mileage	engine	max_power	torque	seats
Min.	1994	509.983	1	0	624	32.8	4.8	2
1st Qu.	2012	4590	35000	16.78	1197	68.05	101	5
Median	2015	7650	60000	19.3	1248	82	154.9	5
Mean	2014	11046.833	69188.66	19.41986	1458.709	91.58737	168.2941	5.416393
3rd Qu.	2017	11730	95425	22.32	1582	102	202	5
Max.	2020	170000	2360457	42	3604	400	789	14

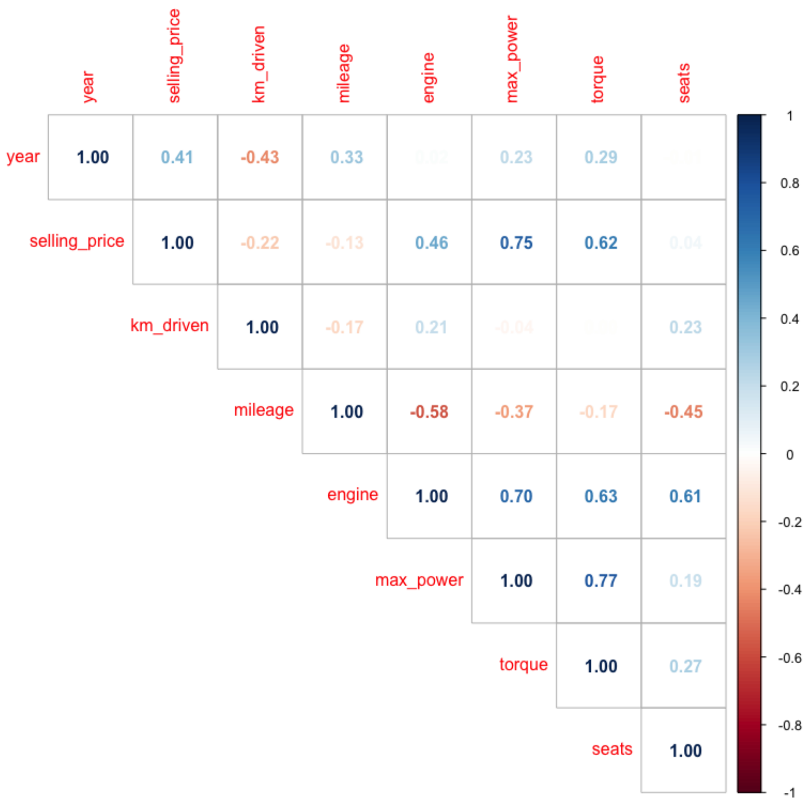
(Table 1. Summary statistics of numerical variables in the data)

fuel	seller_type	transmission	owner
CNG:52	Dealer:1107	Automatic:1041	First Owner:5215
Diesel:4299	Individual:6563	Manual:6865	Fourth & Above Owner:160
LPG:35	Trustmark Dealer:236		Second Owner:2016
Petrol:3520			Test drive car: 5
			Third Owner:510

(Table 2. Summary statistics of categorical variables in the data)

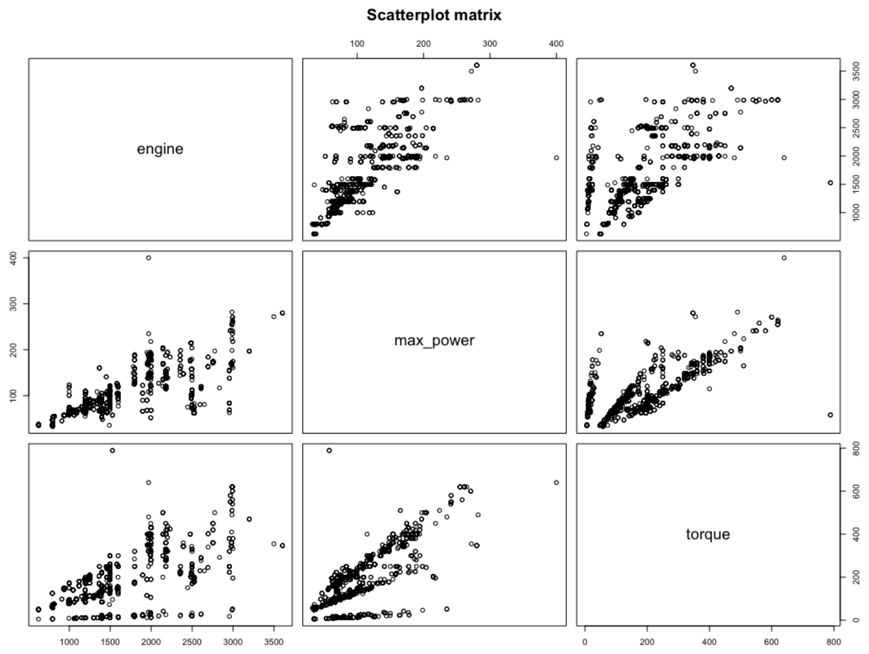
Table 1 showed that our data has a wide range of values, meaning the existence of possible outliers and influential points is viable and they may affect our analysis. For example, the variable Km_driven shows that the minimum value is 1 km while the maximum value is 2360457 km, but most values do not reach the extreme value of 2360457 km.
Table 2 above shows the counts of categorical data in the data set, it is clear that these variables are unbalanced and may cause problems in generalizing the model predictions.

The correlation plot (figure.1) of all the numerical predictors and response variable (Selling Price) below shows the Pearson correlation between each predictors. We are examining the correlation to identify predictors that may cause potential collinearity in our model. From the plot, we can see that the correlation between engine, max power, and torque are high. This is largely due to the fact that they are measures of engine features which depend on one another. Engine variable specifies the gas intake of the car's engine, the max power variable represents the maximum brake horsepower of the car, and torque contains the value of the force generated by the engine. These variables are highly similar in that they all represent certain measures of the engine's power.



(Figure 1. Correlation plot of all numeric variables)

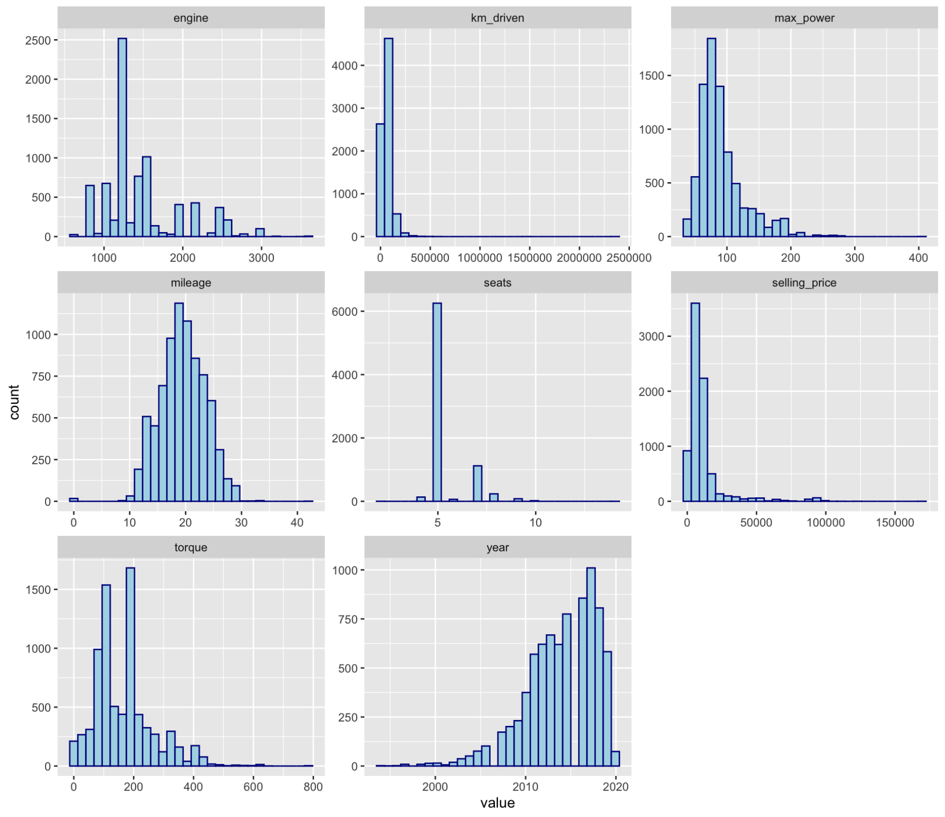
We now take a closer look at these 3 highly correlated variables: engine, max power and torque.



(Figure 2. Scatter plot matrix of the 3 highly correlated variables)

As you can see from the plot above, there is an upward, positive relationship between the variables. The inclusion of these variables may cause multicollinearity to be present in our model, we will need to carefully consider whether to keep these variables in the modelling stages.

Next, we investigate the distribution of the numerical variables in the dataset to identify potential problems. The histogram of the numerical variables shows that variables such as max power, and selling price are skewed, while the variables such as engine are bimodal. These characteristics may cause potential fitting problems when the extreme values are included in the model.



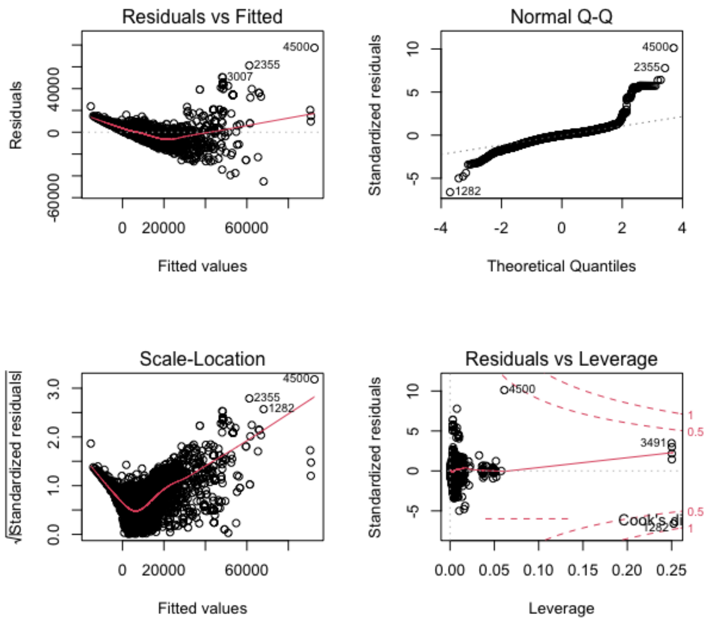
(Figure 3. Histograms of numerical predictors in the dataset)

We ultimately decided to keep these variables as we believe they are strong predictors to our target variable (selling price) and so they will be beneficial to creating our models. With a better understanding of the dataset, we move on to creating models for predictions.

Multi-linear Regression

The MLR model was created as a baseline model to check model assumptions, process the necessary transformations, and remove influential points. The fitted model showed that all three model assumptions were violated (figure.1). Both the Shapiro-Wilk normality test and BP test for equal variance yield p-value smaller than 0.05, hence we conclude that the assumptions are violated (SW: P-V <2.2e16, BP:P-V<2.2e-16). A few transformations were considered: removing influential points, box-cox, quadratic and cubic models. By removing influential points and applying box-cox transformations, the model still did not pass the model assumptions (both methods had p-values smaller than 0.05). Next we use transformations on predictors by adding polynomial terms in the fitted models.

By examining the summary table of the fitted model, we noticed only 5 predictors were insignificant out of the 18. The predictors for different types of fuels (Diesel, LPG, Petrol) had p-values (0.4,0.3,0.2 respectively) greater than 0.2. The predictor for owner type (fourth level and above) as well as owner type (third level) were also insignificant (p-values >0.05). From this simple model, we assume there are strong linear relationships between our response variable and our predictors.

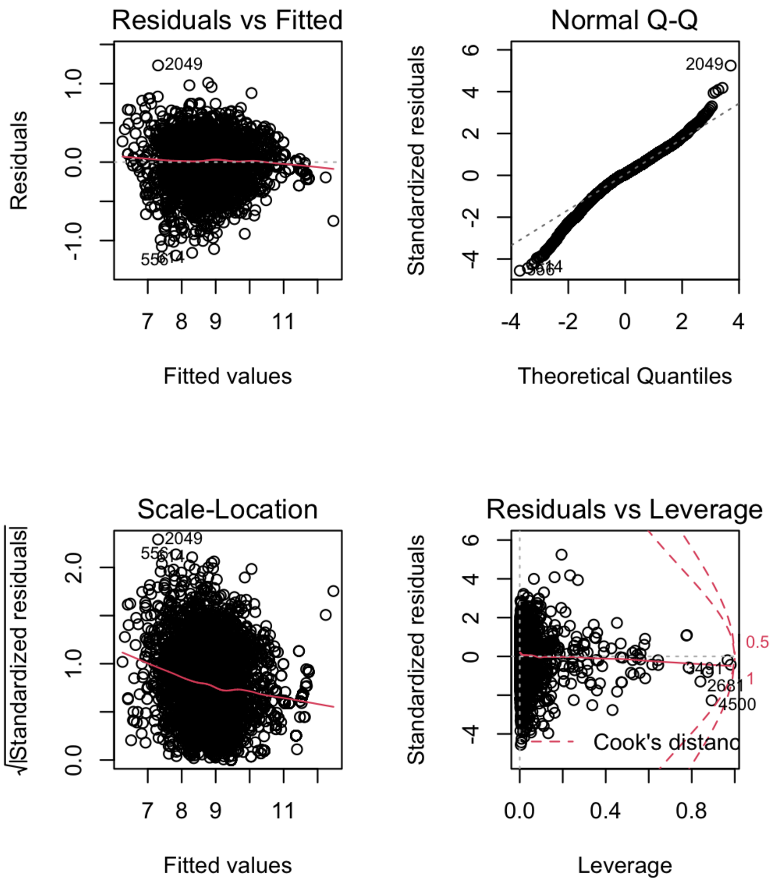


(Figure 4. Model diagnostic charts of non-transformed MLR with all predictors)

Quadratic Model

As the assumptions did not hold with the MLR model, we created a quadratic model in hopes that the model assumptions would not be violated. The quadratic model is as follows,

$$fit\_2 = lm((selling\_price) \sim . + \ .\hat{}2 \ , \ data = car.tr)$$



(Figure 5. Model diagnostic charts of Quadratic model)

As shown above, both the residual plot and the normal QQ plot do not provide any evidence of compliance to the assumptions. Furthermore, on the BP test, the almost trivial P-value (< 2.2e-16) indicates a strong violation of the null assumption, the normality of the residual is also failed for this quadratic model.

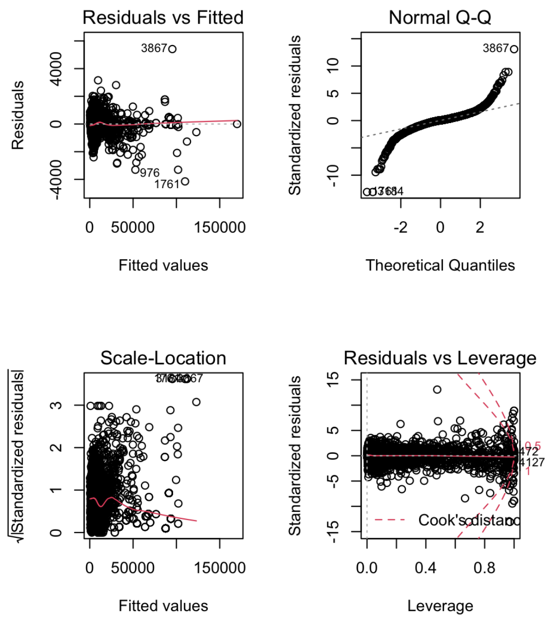
Looking at the residual plot we see that the scattered plots are not evenly distributed in a rectangular area, where the residual variance varies across different fitted values. Therefore we conclude that the equal variance assumption is violated.

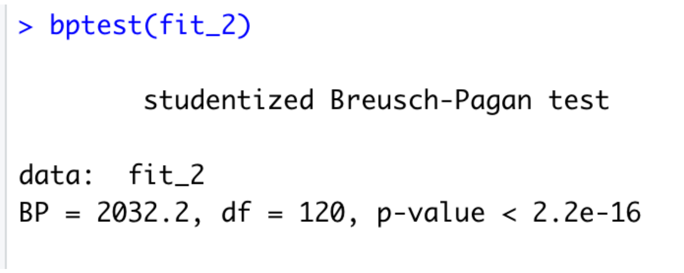
The normal QQ plot shows the x-axis which is the 'theoretical' quantile - these are the 'real' quantiles from a true normal distribution - and the y-axis is the empirical quantile. From the fitness of the QQ plot, one can observe that the empirical quantiles are not well fitted against the theoretical quantiles. Therefore, it is reasonable to conclude that the normality variance is violated.

The Shapiro-Wilk test uses a null hypothesis that the normality condition holds, versus an alternative hypothesis that the normality assumption is violated. As we mentioned, the p-value (p-value < 2.2e-16) is almost trivial, so we reject the null hypothesis and thus conclude the normality is violated.

Quartic Model

Since the model assumptions were violated in the quadratic model, we attempt to create another model, quartic, in hopes of these model assumptions passing.

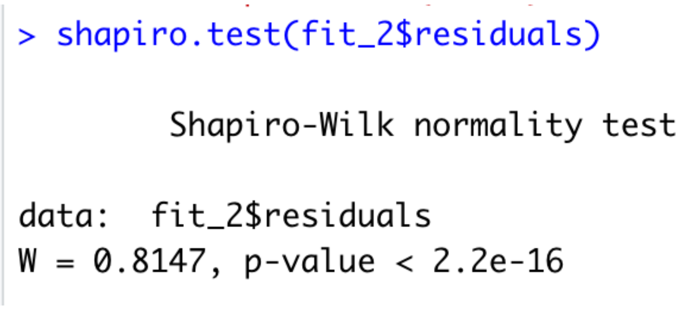




```
> bptest(fit_2)

	studentized Breusch-Pagan test

data:  fit_2
BP = 2032.2, df = 120, p-value < 2.2e-16
```



```
> shapiro.test(fit_2$residuals)

	Shapiro-Wilk normality test

data:  fit_2$residuals
W = 0.8147, p-value < 2.2e-16
```

(Figure 6. Model diagnostic charts of Quartic model and the corresponding BP and Shapiro tests)

Comparing the quadratic QQ plot to the quartic QQ plot, they both fail to follow the line. Although the quartic model seems to follow the line better - as it is closer to the line and has less of a diverging tail. The normality check (via Shapiro tests) shows that the normality assumptions are well satisfied. However, such a statement is simply a rephrase of a graphical observation through naked eye, which is highly intuitive and empirical. Thus more statistically solicidated tests are to be carried out to further confirm or deny such rough observation.

The quartic model shows similar trends compared to the quadratic model. The shrinking pattern in the residual plot, and the significant deviation from the theoretical quantiles in the normal QQ plot, give evidence against the linearity and normality assumptions. Our BP and Shapiro tests both suggest very insignificant almost trivial P-values as shown above. Therefore not all of the theoretical assumptions are in place yet.

Listed below are some metrics we obtained from fitting the training dataset on each model.

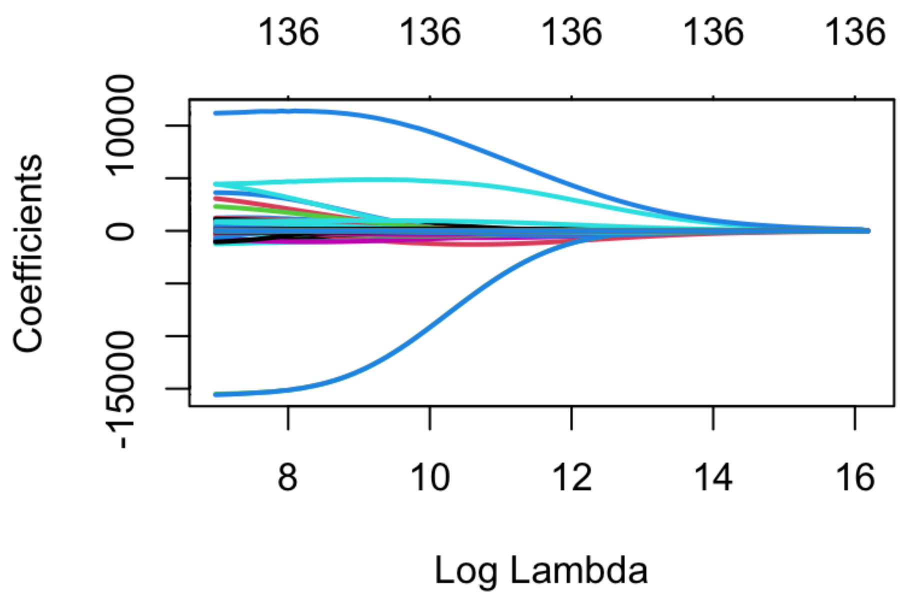
Model	RMSE	R-squared	Adjusted R-squared
Linear	7928.58	0.691	0.69
Quadratic	2340.42	0.919	0.917
Quartic	2942.37	0.98	0.98

(Table 3. Summary statistics of the linear, quadratic and quartic models)

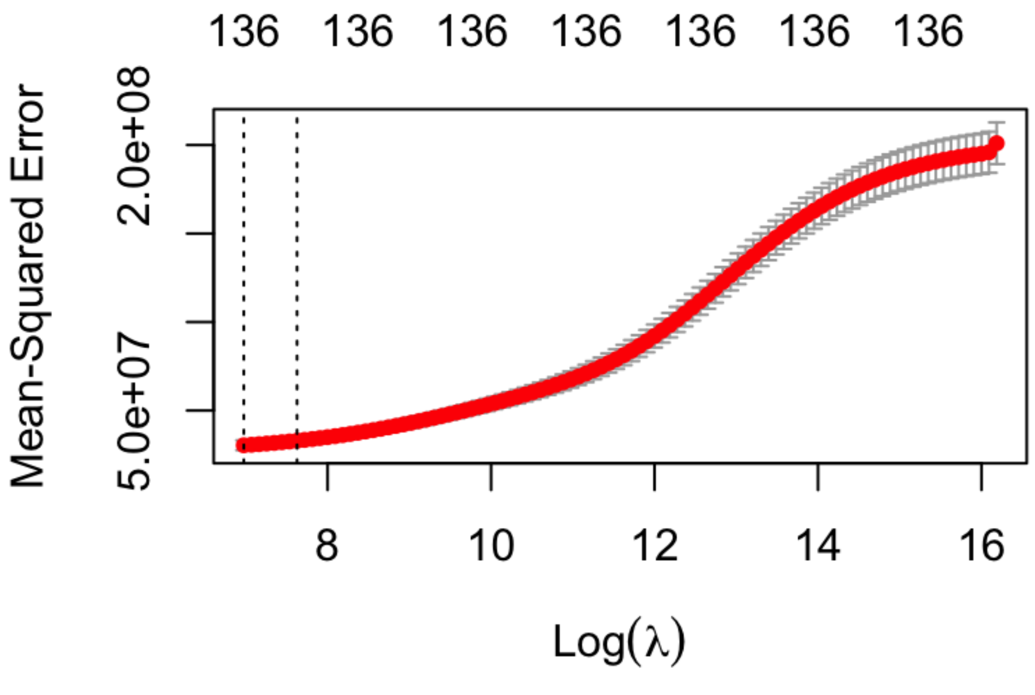
The quadratic model performs the best in both RMSE and R-squared as it has the lowest RMSE value and highest R-squared value. Therefore, we decide to move forward with the quadratic fit to see how it performs in Ridge, Lasso and Elastic net models.

Ridge Regression

Ridge regression is beneficial to use when we need to estimate coefficients of regression models where the independent variables are highly correlated. In this model, the coefficients are reduced so that the variables have coefficients close to zero. Shrinking the coefficients is accomplished by adding a penalty term of L2-norm (sum of the squared coefficients) to the regression model. It is crucial to choose an adequate penalty for your model, as the penalty impacts the shrinkage; when the penalty increases, the shrinkage increases resulting in the coefficients to get closer to zero. This is shown in figure 7. In our dataset, we found that the optimal penalty (lambda λ) was 1069.588 and figure 8 shows the best log(lambda λ). Using this penalty, the coefficients of our variables are shown in the Appendix code. Looking at the coefficients, they differ from one another quite drastically. This model performs the best when the predictors have roughly the same size coefficients. This is one of the reasons why we decided to perform Lasso regression on our data, as this model performs better when the coefficients differ greatly. There is also the disadvantage with Ridge regression as it shrinks the coefficients close to zero, but never sets the coefficients to "exactly" zero. Therefore, we decided to apply Lasso regression to accommodate these problems.



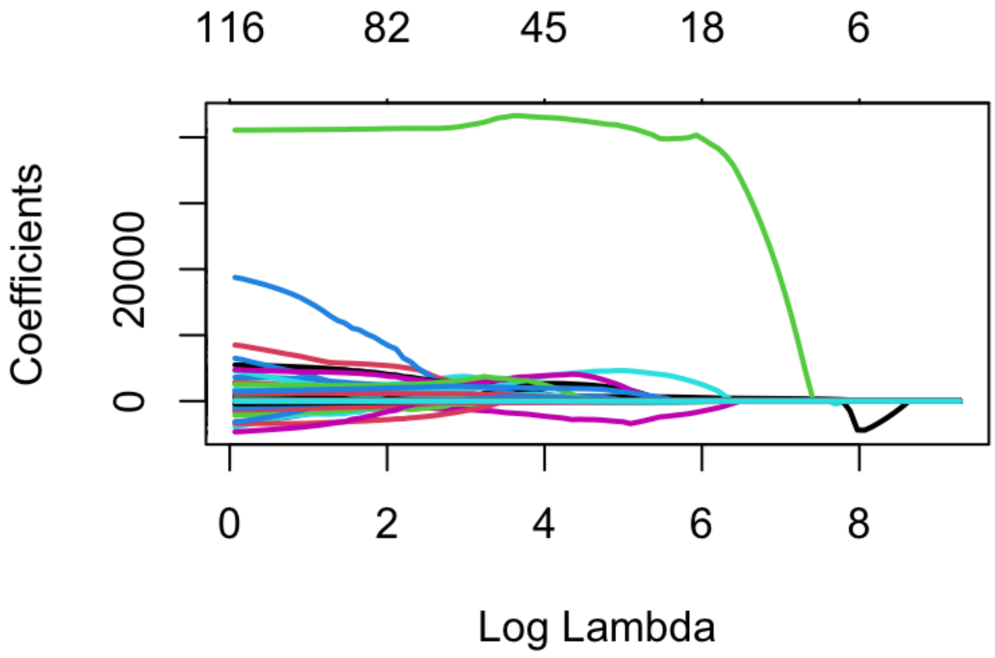
(Figure 7. Graph displaying the shrinkage of coefficients as log lambda (penalty) increases)



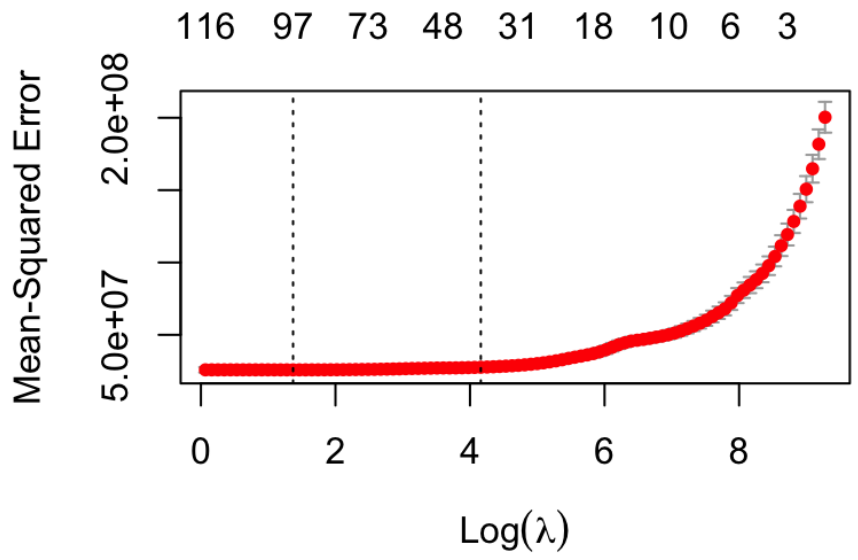
(Figure 8. Graph displaying the best log lambda for the model)

Lasso Regression

In this model, the coefficients are reduced towards zero and unlike Ridge, they can be “equal” to zero. Shrinking the coefficients is accomplished by adding a penalty term of L1-norm (sum of the absolute coefficients) to the regression model. This model reduces complexity which is why it is often an alternative to subset selection methods for performing variable selection. As seen in the Ridge regression coefficients, some of the predictors have large coefficients compared to another. Meaning Lasso regression should perform better for our dataset. Another benefit of Lasso regression is it performs a feature selection automatically which helps prevent overfitting. We found the optimal lambda (penalty) for our dataset to be 3.93435. Using this penalty, the coefficients of our variables are shown in the Appendix code.



(Figure 9. Graph displaying the shrinkage of coefficients as log lambda (penalty) increases)



(Figure 10. Graph displaying the shrinkage of coefficients as log lambda (penalty) increases)

Elastic Net Regression

The Elastic Net regression model combines both penalties of the Lasso and Ridge models, to shrink the coefficients both L1 and L2 norm are used - some of the coefficients are shrunk and some are set to zero. In this model both alpha and lambda parameters need to be tuned to produce optimal results. The optimal parameters are chosen based on which values minimize the cross validation error. Performing a 10 fold cross validation, we found the optimal alpha to be 0.7 and the lambda to be 4.941773. The coefficients of our variables are shown in the Appendix code.

Model Comparison

In each model, their performance metrics were calculated via prediction error (RMSE) and R-squared, where the best performing model would be the one that minimizes RMSE.

Model	**R-Squared (R^2)**	**RMSE**
Ridge	0.8450274	5234.425
Lasso	0.8651846	4883.61
Elastic net	0.8655011	4876.354

(Table 4. Summary statistics of Ridge, Lasso and Elastic net models)

Looking at the median RMSE values, we see that for the Ridge, Lasso and the Elastic net model, the RMSE is 5234.425, 4883.61 and 4876.354 respectively. Since the Elastic net model has the lowest median RMSE, we can conclude that this is the best performing model out of the three.

Stepwise Regression

Stepwise regression uses a step iterative method on a regression model where it selects independent variables to be used in the final model. Depending on whether a forward or backward method is used, it adds or removes variables after each iteration, testing for statistical significance of the variable. In backward selection, the full model is used initially (has all the variables) and after each iteration, the variables that are not significant are dropped one at a time. In forward selection, it starts from the null model (no variables) and adds beneficial variables to the model one at a time until the specified criterion is met. There are benefits to both methods, but we decided to use the backward method as the forward method can produce suppressor effects. Suppressor effects are present when predictors only appear to be significant when another predictor is held constant. As there seems to be collinearity in our model, suppressor effects seem more likely to occur in our model. Therefore, we ultimately decided to use the backward method to avoid this.

We used both AIC and BIC criterions in the stepwise selection to check the significance of each predictor. AIC prefers more complex models which can result in overfitting. BIC prefers a simpler model - aiming to be the true model but it can result in underfitting.

First we look at the AIC stepwise selection model, the start AIC is 72636.84 and the end AIC is 72603.46. By the end there are 91 coefficients, the R-squared is 0.9181 and the adjusted R-squared is 0.9168. The RMSE is 18083.66.

Now we look at the BIC stepwise selection model, the start BIC is 73506.09 and the end BIC is 72992.91. By the end there are 56 coefficients, the R-squared is 0.917 and the adjusted R-squared is 0.9161. The RMSE is 18091.84.

We performed an ANOVA test on both models to see whether they are significantly different. Model 1 is the AIC stepwise selection model and model 2 is the BIC stepwise selection model. The p-value is 9.185e-5 which is less than 0.05, so we conclude that the models are significantly different. Due to this, a further ANOVA test was conducted to determine which model to use. This test proved that at least one of the predictors that were significant did not appear in the BIC stepwise selection model. Since AIC has the lower RMSE value and higher (more fit) R-squared and adjusted R-squared values - as well as being shown that significant predictors were dropped in BIC - we conclude that using the AIC stepwise selection model is more preferable.

Results

Model	R-Squared	Adjusted R-Squared	RMSE
Linear	0.691	0.690	7928.58
Quadratic	0.919	0.917	2340.42
Quartic	0.980	0.980	2942.37
Ridge	0.8450274	0.842088757	5234.425
Lasso	0.8651846	0.862628184	4883.61
Elastic Net	0.8655011	0.862950686	4876.354
AIC	0.9181	0.9168	18083.66
BIC	0.917	0.9161	18091.84

(Table 5. Summary statistics of all models conducted in this report)

The table (Table 5.) above shows the complete model accuracy measure returned by our regression models. We can observe that the quadratic model has the lowest RMSE with very high R-squared and adjusted R-square.

Limitations

Our main source of limitation is due to the limited data we have. Data about the cars market is mainly dominated by some of the main dealers. Unfortunately, these main dealers selectively release data that they decide will be most beneficial to their sales. So, the data the public sees is limited and biased. Also due to the limited amount of data we have, some of the models result to be misleading. After this project, consistently following up with updates on the relevant market will help the n-p relationship be more refined. Some factors that would be beneficial to explore to further improve our prediction on the selling price of used cars would be: sales period for the used vehicle and amount of discount from the original price.

Another significant limitation is that a considerable amount of assumptions and/or conditions are not well satisfied for two of our models. As we conducted the tests on linear, quadratic and quartic models, we found that many presumptions were not present in the data. For example, in the production of the residual and normal QQ plots, it was observed that the three main assumptions: linearity, equal variance, and normality in the residuals. In our check for these assumptions, we realized that the dataset is not perfectly theoretically fitted for the linear, quadratic and quartic models, but their suitability could theoretically be improved by conducting a set of variable transformations, relevant tools including Box-Cox method, logarithm and variable selections (AIC/BIC, backward/forward/stepwise), and applying penalties (Ridge, LASSO, Elastic net). All these methods produced similar results, unfortunately not satisfying the model assumptions. Therefore we conclude that in practice, it would be more beneficial to use models that are more relevant to the data (Random Forest, XGboosting, etc.) would be more beneficial in trying to predict the trend.

With regards to the quadratic model, we can see the values of R-square measure and adjusted R-square are both high with low RMSE measure. Therefore, there is a possibility that our model is overfitting which may cause the model to perform worse when it is introduced to new data.

Concluding thoughts

The main purpose of our research is to find and create a model that can accurately predict the selling price of a used car based on its specific features that are not affected by economical factors. We used the Linear Regression model in this report as it is one of the most traditional and widely used models. We fitted different forms of Linear Regressions including basic Multi-linear Regression, Quadratic Linear Regression, Quartic Linear Regression, Ridge Regression, Lasso Regression and Elastic net regression.

The initial model using MLR showed that multiple variables were highly significant, which carried over to the following fitted models after. Although the model assumptions were violated, it was clear that there existed some highly significant relationships between our response variable "Selling price" and the predictors. In order to make accurate predictions we attempted to transform the data by applying Box-Cox, log and polynomial transformations. However, the results were not ideal as the model assumptions were still violated.

We continued to explore different options with modelling selling price and the predictors by implementing Ridge, Lasso and Elastic Net models. In the end, it was found that these models performed worse than the quadratic model, as they all had lower RMSE and R-squared values.

Looking at all models, we concluded that the quadratic model performed the best for our dataset because it had the lowest RMSE with a relatively high R-squared value. This model explained 91.9% of the variation in the selling price leaving 8.1% unexplained. We looked further into the significance and the importance of each predictors' effect on the response variable in the quadratic model. Appendix 1 shows these coefficients were significant in the quadratic model. These coefficients show us how much we need to adjust the trajectory of our predicted selling price. For example the coefficient for km_driven is 4.264, that means for every 4.264 km driven, the selling price goes up by one dollar. The quadratic model uses all these coefficients as factors to predict the selling price, with these coefficients, a mathematical formula can be made that calculates the selling price. The formula would multiply the coefficients with the values we know about the car to get the predicted selling price.

In terms of regularization, we found that Lasso and Elastic net are a good tie, as they had very similar values for adjusted R-squared and RMSE. While Ridge regularization is less optimal than its two counterparties due to its higher RMSE and lower R-squared.

For variable selection, both stepwise AIC and BIC gave approximately equivalent performance, so none of these two significantly overwhelmed the other. In this case, we decided the eventual prediction performance is indifferent to the selection of penalty for model complexity.

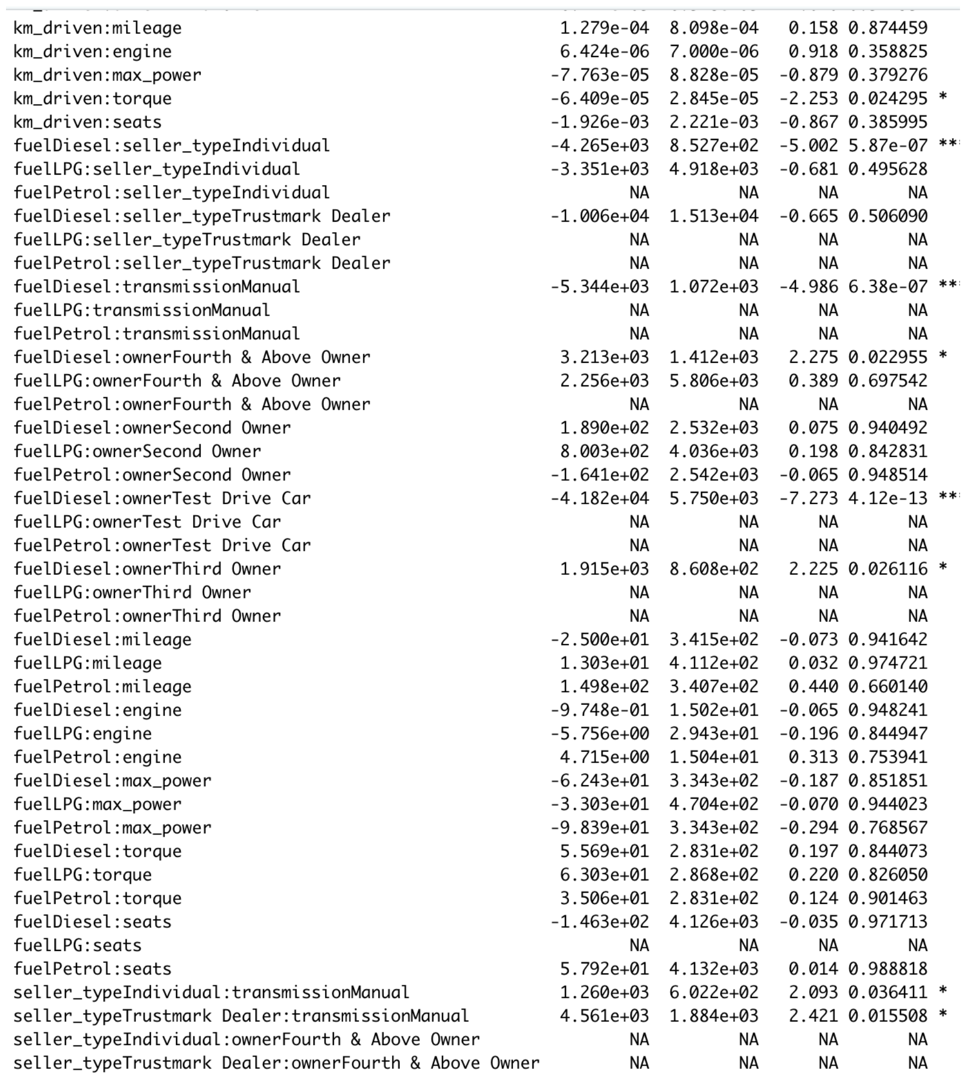
References

1. N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), 2018, pp. 115-119, doi: 10.1109/ICBIR.2018.8391177.
2. Engers, M., Hartmann, M., & Stern, S. (2009). Annual miles drive used car prices. *Journal of Applied Econometrics*, *24*(1), 1-33.

Appendix

Appendix 1. Coefficients of the Quadratic Model

```
                                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)                            -1.297e+06  1.087e+06  -1.193 0.232768    
year                                    6.312e+02  5.388e+02   1.171 0.241479    
km_driven                               4.264e+00  1.056e+00   4.038 5.47e-05 ***
fuelDiesel                              3.064e+04  8.973e+05   0.034 0.972759    
fuelLPG                                 1.886e+05  1.665e+06   0.113 0.909802    
fuelPetrol                              5.105e+05  8.917e+05   0.573 0.566984    
seller_typeIndividual                   6.209e+05  1.817e+05   3.416 0.000640 ***
seller_typeTrustmark Dealer            -3.635e+05  9.482e+05  -0.383 0.701501    
transmissionManual                      2.326e+06  2.423e+05   9.600  < 2e-16 ***
ownerFourth & Above Owner              -7.111e+04  2.754e+05  -0.258 0.796241    
ownerSecond Owner                       7.097e+04  1.109e+05   0.640 0.522144    
ownerTest Drive Car                     3.669e+04  5.607e+03   6.545 6.61e-11 ***
ownerThird Owner                        1.187e+05  1.822e+05   0.652 0.514559    
mileage                                -3.962e+04  1.625e+04  -2.439 0.014773 *  
engine                                  1.355e+02  2.170e+02   0.625 0.532147    
max_power                              -3.555e+04  2.649e+03 -13.420  < 2e-16 ***
torque                                 -3.972e+03  9.425e+02  -4.215 2.55e-05 ***
seats                                   4.691e+04  6.420e+04   0.731 0.464992    
year:km_driven                         -2.148e-03  5.279e-04  -4.068 4.81e-05 ***
year:fuelDiesel                        -9.818e+00  4.438e+02  -0.022 0.982353    
year:fuelLPG                           -8.959e+01  8.252e+02  -0.109 0.913548    
year:fuelPetrol                        -2.558e+02  4.411e+02  -0.580 0.561992    
year:seller_typeIndividual             -3.058e+02  9.050e+01  -3.379 0.000734 ***
year:seller_typeTrustmark Dealer        1.640e+02  4.688e+02   0.350 0.726484    
year:transmissionManual                -1.152e+03  1.206e+02  -9.554  < 2e-16 ***
year:ownerFourth & Above Owner          3.828e+01  1.378e+02   0.278 0.781153    
year:ownerSecond Owner                 -3.607e+01  5.553e+01  -0.650 0.515953    
year:ownerTest Drive Car                       NA         NA      NA       NA    
year:ownerThird Owner                  -4.961e+01  9.140e+01  -0.543 0.587301    
year:mileage                            1.974e+01  8.058e+00   2.450 0.014332 *  
year:engine                            -7.315e-02  1.075e-01  -0.680 0.496339    
year:max_power                          1.790e+01  1.304e+00  13.723  < 2e-16 ***
year:torque                             1.939e+00  4.464e-01   4.344 1.43e-05 ***
year:seats                             -2.181e+01  3.187e+01  -0.684 0.493933    
km_driven:fuelDiesel                    4.060e-03  2.010e-02   0.202 0.839915    
km_driven:fuelLPG                      -2.446e-02  5.355e-02  -0.457 0.647888    
km_driven:fuelPetrol                   -2.043e-04  1.997e-02  -0.010 0.991840    
km_driven:seller_typeIndividual         3.636e-02  7.701e-03   4.722 2.40e-06 ***
km_driven:seller_typeTrustmark Dealer   9.164e-02  4.925e-02   1.861 0.062871 .  
km_driven:transmissionManual            3.729e-02  8.307e-03   4.489 7.33e-06 ***
km_driven:ownerFourth & Above Owner    -7.353e-03  1.121e-02  -0.656 0.511943    
km_driven:ownerSecond Owner            -4.606e-03  3.916e-03  -1.176 0.239584    
km_driven:ownerTest Drive Car           5.552e-01  3.505e-01   1.584 0.113304    
km_driven:ownerThird Owner             -6.412e-03  6.348e-03  -1.010 0.312534    
```



```
km_driven:mileage                                  1.279e-04  8.098e-04   0.158 0.874459
km_driven:engine                                   6.424e-06  7.000e-06   0.918 0.358825
km_driven:max_power                               -7.763e-05  8.828e-05  -0.879 0.379276
km_driven:torque                                  -6.409e-05  2.845e-05  -2.253 0.024295 *
km_driven:seats                                   -1.926e-03  2.221e-03  -0.867 0.385995
fuelDiesel:seller_typeIndividual                  -4.265e+03  8.527e+02  -5.002 5.87e-07 **
fuelLPG:seller_typeIndividual                     -3.351e+03  4.918e+03  -0.681 0.495628
fuelPetrol:seller_typeIndividual                          NA         NA      NA       NA
fuelDiesel:seller_typeTrustmark Dealer            -1.006e+04  1.513e+04  -0.665 0.506090
fuelLPG:seller_typeTrustmark Dealer                       NA         NA      NA       NA
fuelPetrol:seller_typeTrustmark Dealer                    NA         NA      NA       NA
fuelDiesel:transmissionManual                     -5.344e+03  1.072e+03  -4.986 6.38e-07 **
fuelLPG:transmissionManual                                NA         NA      NA       NA
fuelPetrol:transmissionManual                             NA         NA      NA       NA
fuelDiesel:ownerFourth & Above Owner               3.213e+03  1.412e+03   2.275 0.022955 *
fuelLPG:ownerFourth & Above Owner                  2.256e+03  5.806e+03   0.389 0.697542
fuelPetrol:ownerFourth & Above Owner                      NA         NA      NA       NA
fuelDiesel:ownerSecond Owner                       1.890e+02  2.532e+03   0.075 0.940492
fuelLPG:ownerSecond Owner                          8.003e+02  4.036e+03   0.198 0.842831
fuelPetrol:ownerSecond Owner                      -1.641e+02  2.542e+03  -0.065 0.948514
fuelDiesel:ownerTest Drive Car                    -4.182e+04  5.750e+03  -7.273 4.12e-13 **
fuelLPG:ownerTest Drive Car                               NA         NA      NA       NA
fuelPetrol:ownerTest Drive Car                            NA         NA      NA       NA
fuelDiesel:ownerThird Owner                        1.915e+03  8.608e+02   2.225 0.026116 *
fuelLPG:ownerThird Owner                                  NA         NA      NA       NA
fuelPetrol:ownerThird Owner                               NA         NA      NA       NA
fuelDiesel:mileage                                -2.500e+01  3.415e+02  -0.073 0.941642
fuelLPG:mileage                                    1.303e+01  4.112e+02   0.032 0.974721
fuelPetrol:mileage                                 1.498e+02  3.407e+02   0.440 0.660140
fuelDiesel:engine                                 -9.748e-01  1.502e+01  -0.065 0.948241
fuelLPG:engine                                    -5.756e+00  2.943e+01  -0.196 0.844947
fuelPetrol:engine                                  4.715e+00  1.504e+01   0.313 0.753941
fuelDiesel:max_power                              -6.243e+01  3.343e+02  -0.187 0.851851
fuelLPG:max_power                                 -3.303e+01  4.704e+02  -0.070 0.944023
fuelPetrol:max_power                              -9.839e+01  3.343e+02  -0.294 0.768567
fuelDiesel:torque                                  5.569e+01  2.831e+02   0.197 0.844073
fuelLPG:torque                                     6.303e+01  2.868e+02   0.220 0.826050
fuelPetrol:torque                                  3.506e+01  2.831e+02   0.124 0.901463
fuelDiesel:seats                                  -1.463e+02  4.126e+03  -0.035 0.971713
fuelLPG:seats                                             NA         NA      NA       NA
fuelPetrol:seats                                   5.792e+01  4.132e+03   0.014 0.988818
seller_typeIndividual:transmissionManual           1.260e+03  6.022e+02   2.093 0.036411 *
seller_typeTrustmark Dealer:transmissionManual     4.561e+03  1.884e+03   2.421 0.015508 *
seller_typeIndividual:ownerFourth & Above Owner           NA         NA      NA       NA
seller_typeTrustmark Dealer:ownerFourth & Above Owner     NA         NA      NA       NA
```

```
seller_typeIndividual:ownerSecond Owner              7.827e+02  6.254e+02   1.251 0.210860    
seller_typeTrustmark Dealer:ownerSecond Owner       -1.780e+03  3.268e+03  -0.545 0.586011    
seller_typeIndividual:ownerTest Drive Car                   NA         NA      NA       NA    
seller_typeTrustmark Dealer:ownerTest Drive Car             NA         NA      NA       NA    
seller_typeIndividual:ownerThird Owner              -5.170e+03  2.982e+03  -1.734 0.083002 .  
seller_typeTrustmark Dealer:ownerThird Owner                NA         NA      NA       NA    
seller_typeIndividual:mileage                       -8.988e+01  8.200e+01  -1.096 0.273081    
seller_typeTrustmark Dealer:mileage                  2.952e+02  2.933e+02   1.007 0.314201    
seller_typeIndividual:engine                         1.170e+00  1.060e+00   1.103 0.270063    
seller_typeTrustmark Dealer:engine                   2.415e+00  1.196e+01   0.202 0.839984    
seller_typeIndividual:max_power                     -1.386e+02  1.469e+01  -9.433  < 2e-16 ***
seller_typeTrustmark Dealer:max_power               -5.746e+01  1.058e+02  -0.543 0.587217    
seller_typeIndividual:torque                         3.547e+01  5.831e+00   6.082 1.28e-09 ***
seller_typeTrustmark Dealer:torque                   6.992e+01  1.678e+02   0.417 0.677025    
seller_typeIndividual:seats                          1.252e+02  3.433e+02   0.365 0.715360    
seller_typeTrustmark Dealer:seats                    2.692e+03  8.790e+03   0.306 0.759369    
transmissionManual:ownerFourth & Above Owner         2.493e+02  2.573e+03   0.097 0.922818    
transmissionManual:ownerSecond Owner                -3.193e+02  7.283e+02  -0.438 0.661052    
transmissionManual:ownerTest Drive Car                      NA         NA      NA       NA    
transmissionManual:ownerThird Owner                 -9.067e+03  1.799e+03  -5.040 4.83e-07 ***
transmissionManual:mileage                           2.325e+02  1.116e+02   2.083 0.037316 *  
transmissionManual:engine                            1.544e+01  1.453e+00  10.623  < 2e-16 ***
transmissionManual:max_power                        -3.192e+02  1.710e+01 -18.665  < 2e-16 ***
transmissionManual:torque                            1.753e+01  6.817e+00   2.572 0.010156 *  
transmissionManual:seats                            -7.378e+02  4.724e+02  -1.562 0.118420    
ownerFourth & Above Owner:mileage                   -2.023e+02  1.675e+02  -1.208 0.227095    
ownerSecond Owner:mileage                            7.965e+01  7.068e+01   1.127 0.259855    
ownerTest Drive Car:mileage                                 NA         NA      NA       NA    
ownerThird Owner:mileage                            -2.205e+02  1.269e+02  -1.737 0.082496 .  
ownerFourth & Above Owner:engine                    -2.542e+00  2.020e+00  -1.259 0.208129    
ownerSecond Owner:engine                             3.653e-01  7.187e-01   0.508 0.611295    
ownerTest Drive Car:engine                                  NA         NA      NA       NA    
ownerThird Owner:engine                             -2.183e+00  1.308e+00  -1.669 0.095113 .  
ownerFourth & Above Owner:max_power                  3.296e+01  2.811e+01   1.173 0.241007    
ownerSecond Owner:max_power                         -9.350e+00  9.820e+00  -0.952 0.341105    
ownerTest Drive Car:max_power                               NA         NA      NA       NA    
ownerThird Owner:max_power                          -3.866e+00  1.590e+01  -0.243 0.807947    
ownerFourth & Above Owner:torque                    -2.793e+00  6.231e+00  -0.448 0.653920    
ownerSecond Owner:torque                             8.817e-01  3.477e+00   0.254 0.799824    
ownerTest Drive Car:torque                                  NA         NA      NA       NA    
ownerThird Owner:torque                              4.540e+00  4.437e+00   1.023 0.306311    
ownerFourth & Above Owner:seats                     -5.590e+02  5.608e+02  -0.997 0.318941    
ownerSecond Owner:seats                             -3.774e+01  2.398e+02  -0.157 0.874943    
ownerTest Drive Car:seats                                   NA         NA      NA       NA    
ownerThird Owner:seats                               1.890e+02  3.986e+02   0.474 0.635389    
```

```
mileage:engine                                      -4.289e-01  7.177e-02  -5.976 2.47e-09 ***
mileage:max_power                                    7.018e+00  1.031e+00   6.804 1.15e-11 ***
mileage:torque                                      -6.290e-01  5.130e-01  -1.226 0.220187    
mileage:seats                                       -3.152e+01  3.257e+01  -0.968 0.333325    
engine:max_power                                     3.992e-02  1.032e-02   3.868 0.000111 ***
engine:torque                                        4.862e-03  6.117e-03   0.795 0.426801    
engine:seats                                        -3.831e-02  2.968e-01  -0.129 0.897309    
max_power:torque                                    -9.687e-02  4.788e-02  -2.023 0.043134 *  
max_power:seats                                     -1.043e+01  4.760e+00  -2.190 0.028542 *  
torque:seats                                        -2.548e+00  2.033e+00  -1.253 0.210243    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4095 on 4622 degrees of freedom
Multiple R-squared:  0.9188,	Adjusted R-squared:  0.9167 
F-statistic: 435.7 on 120 and 4622 DF,  p-value: < 2.2e-16
```

R codes:

```
# Car data set from kaggle


## read in file
library(tidyverse)
Carv3 <- read_csv("/Users/gelareh/Desktop/Car details v3.csv")
## EDA

#check dimensions of the data set

dim(Carv3)
head(Carv3)


############################data cleaning
#checking for null value
table(is.na(Carv3))

#remove null values
carsv3 = na.omit(Carv3)

#check for dim of carsv3
dim(carsv3)

#lets check which columns are of interests for our study
#we only want to find the physical characteristics of cars that determines the selling price
#hence we do not remove things such as brand and car models that has a economical factors to
it

head(carsv3)
carsv3 = carsv3[,-1]
dim(carsv3)
head(carsv3)

#take the first number set of torque, engine, mileage max power
for (i in 1:length(carsv3$torque)) {
carsv3$torque[i] = str_extract_all(carsv3$torque, "\\d+\\.*+\\d*+")[[i]][1]
carsv3$mileage[i] = str_extract_all(carsv3$mileage, "\\d+\\.*+\\d*+")[[i]]
carsv3$engine[i] = str_extract_all(carsv3$engine, "\\d+\\.*+\\d*+")[[i]]
carsv3$max_power[i] = str_extract_all(carsv3$max_power, "\\d+\\.*+\\d*+")[[i]]
}

dim(carsv3)
```

```
head(carsv3)

#Lets convert the selling price from rupees to CAD
carsv3$selling_price = carsv3$selling_price * 0.017

#data save point
cc = carsv3
#fail save
carsv3 = cc
#convert all numbers variable into numeric values
carsv3[,8:12] = sapply(carsv3[,8:12],as.numeric)
head(carsv3)


#convert some variables into factors
#carsv3$year = as.character(carsv3$year)
carsv3[sapply(carsv3, is.character)] <- lapply(carsv3[sapply(carsv3, is.character)],as.factor)
#carsv3[,4:7] = sapply(carsv3[,4:7],factor)
levels(as.factor(carsv3$owner))
str(carsv3)
#Check all levels
sapply(carsv3[,c(4:7)], levels)




#The pairwise panel graph of numerical features/predictors
library(psych)
pairs.panels(carsv3,
        method = "pearson", # correlation method
        hist.col = "#00AFBB",
        density = TRUE,  # show density plots
        ellipses = F # show correlation ellipses
)

#lets look at the summary statistics
summary(carsv3[,4:7])
df1_summary<-as.data.frame(apply(carsv3[,-4:-7],2,summary))
df1_summary
#write_csv(df1_summary, file = "/Users/andyc/Desktop/Masters/MDA 9159 Stat
modelling/Project/df1_summary.csv")

df2_summary<-(summary(carsv3[,4:7]))
```

```
(df2_summary)

#correlation plots
library(corrplot)
M <- cor(carsv3[,-4:-7])
corrplot(M, method="number", type = "upper")

#scatter plots of the three variables
pairs(engine ~ max_power+torque, carsv3 , main="Scatterplot matrix")

#histograms of all numeric variables
carsv3 %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value),color = value) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram(color="darkblue", fill="lightblue")

#density plot of all numeric variables
carsv3 %>%
  keep(is.numeric) %>%                    # Keep only numeric columns
  gather() %>%                             # Convert to key-value pairs
  ggplot(aes((value))) +                   # Plot the values
  facet_wrap(~ key, scales = "free") +   # In separate panels
  geom_density()                          # as density

# lets split the data in to training and testing
set.seed(10)
n = nrow( carsv3 )
idx = sample(n, n*0.6, replace = F)
car.tr = carsv3[idx,]
car.ts = carsv3[-idx,]

dim(carsv3);dim(car.tr);dim(car.ts)

# lets fit a linear regression
lm1 = lm(selling_price~. ,data = car.tr)
summary(lm1)

#lets check model diagnostics
par(mfrow=c(2,2))
plot(lm1)

#lm1$residuals
```

```
library(lmtest)

bptest(lm1);
shapiro.test(lm1$residuals)
#test confirms that EV is violated and Normality is violated


#lets try removing the outliers  points
# The obs that are influential point
influ=which(cooks.distance(lm1) > 4/length(cooks.distance(lm1)))
influ
outliers= which((abs(rstandard(lm1)) > 2) & (cooks.distance(lm1) > 4 /
length(cooks.distance(lm1))))
outliers
#remove the outliers
car.nout = carsv3[-outliers,]
n = nrow( car.nout )
idx = sample(n, n*0.6, replace = F)
car.trO = car.nout[idx,]
car.tsO = car.nout[-idx,]
dim(car.nout);dim(car.trO);dim(car.tsO)
#refiit model 1
# lets fit a linear regression
lm2 = lm(selling_price~. ,data = car.tr)
summary(lm2)

#lets check model diagnostics
par(mfrow=c(2,2))
plot(lm2)
lm1$residuals
library(lmtest)

bptest(lm2);
shapiro.test(lm2$residuals)
#test confirms that EV is violated and Normality is violated


#lets do some boxcox transfomation to see if the model assumption is corrected
library(MASS)
graphics.off()
boxcox(lm1,lambda = seq(-0.05, 0.01, by = 0.001))

#The Lambda value is close enough to 0 we can just use log but lets try and see the result
lambda = -0.005
```

```
lm3 <- lm(((car.tr$selling_price^(lambda)-1)/(lambda)) ~ ., data = car.tr)
summary(lm3)
par(mfrow=c(2,2))
plot(lm3)


bptest(lm3);
shapiro.test(lm3$residuals)
#test confirms that EV is violated and Normality is violated


#Sean
################################################################################
#########


# Model comparison (linear/quadratic/quartic)
library(tidyverse)
library(caret)
train.control <- trainControl(method = "cv", number = 10) # Prepares for cv scoring

fit_0 = lm(selling_price ~ 1, data = car.tr) # intercept
fit_1 = lm(selling_price ~ ., data = car.tr) # linear
bptest(fit_1) #BPtest
shapiro.test(fit_1$residuals) #shapiro

par(mfrow=c(2 ,2))
plot(fit_1)
fit_2 = lm(selling_price ~ . + .^2 , data = car.tr) # quadratic
bptest(fit_2) #BPTEST
shapiro.test(fit_2$residuals) #shapiro

par(mfrow=c(2 ,2))
plot(fit_2)
fit_4 = lm(selling_price ~ . +.^2 +.^3 +.^4, data = car.tr)  # quartic
bptest(fit_4) #BPTEST
shapiro.test(fit_4$residuals)#shapiro
par(mfrow=c(2 ,2))
plot(fit_4)

#lets see the model fit performances
library(broom)
print(glance(fit_0)[c(1,2)])
(sqrt(mean(car.ts$selling_price - predict(fit_0, car.ts))^2))
```

```
print(glance(fit_1)[c(1,2)])
(sqrt(mean(car.ts$selling_price - predict(fit_1, car.ts))^2))
print(glance(fit_2)[c(1,2)])
(sqrt(mean(car.ts$selling_price - predict(fit_2, car.ts))^2))
print(glance(fit_4)[c(1,2)])
(sqrt(mean(car.ts$selling_price - predict(fit_4, car.ts))^2))


# The quadratic model has the smallest RMSE and highest Rsquared.
# We apply the quadratic model on the test data:
ypred = predict(fit_2, car.ts)

ytest = car.ts$selling_price
(mse = mean((ypred-ytest)^2))
(rss = sum((ypred-ytest)^2))
(tss = sum((ytest-mean(ytest))^2))

rsq =1-rss/tss
rsq
summary(fit_2)

#save fit_2 coefficients to a csv file
#Df = coef(fit_2)
#write_csv(Df, file ="/Users/gelareh/Desktop/coef.csv")

#Gelareh
################################################################################
##########


#RIDGE, LASSO, ELASTIC NET MODELS
#install.packages("glmnet", dependencies=TRUE)
library(glmnet)

#install.packages("caret", dependencies = TRUE)
library(caret)
set.seed(10)
#creates ridge model
ridge = train(
  selling_price ~ . + .^2, car.tr, method = "glmnet",
  trControl = trainControl("cv", number = 20),
  tuneGrid = expand.grid(alpha = 0, lambda = lambda)
)
```

```
x = model.matrix(selling_price ~ . + .^2, car.tr)[,-1]
y = car.tr$selling_price

#lambda calculation
cv = cv.glmnet(x, y, alpha = 0)
cv$lambda.min

#plot ridge lambda
fit_ridge = glmnet(x, y, alpha = 0)

par(mfrow = c(1, 1))
plot(fit_ridge, xvar = "lambda", label = TRUE,lwd=2)

#plot log lambda
plot(cv)
#visualize the lambda on betas
bestlam = cv$lambda.min
bestlam
log(bestlam)

plot(fit_ridge, xvar = "lambda", label = TRUE,lwd=2)
abline(v=log(bestlam))
#coefficients
coef(ridge$finalModel, ridge$bestTune$lambda)
length(coef(ridge$finalModel, ridge$bestTune$lambda))
#predictions
pred = ridge %>% predict(car.ts)

#model performance metrics
data.frame(
  RMSE = RMSE(pred, car.ts$selling_price),
  Rsquare = R2(pred, car.ts$selling_price)
)

#create lasso model
lasso = train(
  selling_price ~ . + .^2 , car.tr, method = "glmnet",
  trControl = trainControl("cv", number = 20),
  tuneGrid = expand.grid(alpha = 1, lambda = lambda)
)

#lambda calculation
cv = cv.glmnet(x, y, alpha = 1)
```

```
cv$lambda.min
#plot lasso beta conversions
fit_lasso = glmnet(x, y, alpha = 1)
par(mfrow = c(1, 1))
plot(fit_lasso, xvar = "lambda", label = TRUE,lwd=2)
#plot log lambda
plot(cv)
#visualize the lambda on betas
bestlam = cv$lambda.min
bestlam
log(bestlam)

plot(fit_lasso, xvar = "lambda", label = TRUE,lwd=2)
abline(v=log(bestlam))
#coefficients
coef(lasso$finalModel, lasso$bestTune$lambda)
length(coef(lasso$finalModel, lasso$bestTune$lambda))
#predictions
pred = lasso %>% predict(car.ts)

#model performance metrics
data.frame(
  RMSE = RMSE(pred, car.ts$selling_price),
  Rsquare = R2(pred, car.ts$selling_price)
)

#create elastic net model
elastic = train(
  selling_price ~ . + .^2,car.tr, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneLength = 10
)

#lambda values
elastic$bestTune

#coefficients
coef(elastic$finalModel, elastic$bestTune$lambda)
length(coef(elastic$finalModel, elastic$bestTune$lambda))
#predictions
pred = elastic %>% predict(car.ts)

#model performance metrics
data.frame(
  RMSE = RMSE(pred, car.ts$selling_price),
  Rsquare = R2(pred, car.ts$selling_price)
)

#STEP
# backward selection with AIC
# Note that this function uses either AIC or BIC (default: AIC where k=2)
fit_2 = lm(selling_price ~ . + .^2 , data = car.tr)  # quadratic

fit_backwardaic = step(fit_2,direction="backward")
length(fit_backwardaic$coefficients) # = #(beta)
summary(fit_backwardaic)

# Result based on BIC
# Change k (coefficient of the penalty term) to log(n) (default was k=2 (AIC))
fit_back_bic = step(fit_2, direction = "backward", k=log(n))
fit_back_bic
summary(fit_back_bic)
length(fit_back_bic$coefficients) # = #(beta)

#prediction for AIC backward
pred_backward = predict(fit_backwardaic, newdata=carsv3[-idx,])
mse_backward = mean((pred_backward-ytest)^2) # mse for the test data
sqrt(mse_backward)

#prediction for BIC backward
pred_backwardbic = predict(fit_back_bic, newdata=carsv3[-idx,])
mse_backwardbic = mean((pred_backwardbic-ytest)^2) # mse for the test data
sqrt(mse_backwardbic)

#anova tests
anova(fit_backwardaic,fit_back_bic)
anova(fit_back_bic)
anova(fit_backwardaic)
```