

Prediction of car prices regarding their characterisitcs

Hugo Jiménez, Jaume Martínez

April 2020

1 Introduction

In this project we are going to focus on predicting car prices given their characteristics. Our goals are being able to predict the prices with a low error, in order to fit the model as much as possible.

To do so, we have developed a model regression that given some explanatory variables returns an estimation of the price. Doing that, requires some previous steps:

- **Preprocessing** That is the first step which consists in manipulating the data in order to make them more suitable for the following steps.
- **Data visualization** We are going to plot all the variables for checking that all of them fulfill some requirements.
- **Resampling** Some models will be chosen and then compared by using cross-validation. Finally the one with the best performance will be taken.
- **Performance of the model** The model chosen will be tested with the test data and report some analytics.

We realized we had some limitations, for instance, we do not have all variables that explain the price like deterioration of the car, that affects notably to its value. So, the model will not be perfect and will have a considerable error.

Contents

1	Introduction	2
2	Data exploration	4
2.1	Preprocessing	4
2.2	Data visualization	6
3	Resampling	12
3.1	GLM, Lasso Regression and Ridge Regression	13
3.2	Neuronal Network: MLP	13
3.3	Random Forest	13
3.4	Cross Validation	14
4	Results	14
5	Conclusions	15
6	Extensions and limitations	15
7	Appendix	16

2 Data exploration

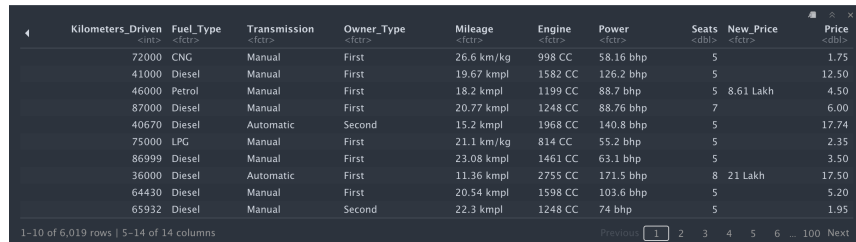
We have obtained the data in the kaggle page, where they provided us two data sets: train-data and test-data. However, the test-data did not have the price variable, so we are not able to compare results. As a solution, we have partitioned the train-data into train and test, but we have done it after preprocessing so all data is treated the same way.

By taking a first look of the data set we have:



X	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic
5	Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual
6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual
7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic
8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual
9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual

Figure 1: Train data set



Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5		1.75
41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5		12.50
46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5	8.61 Lakh	4.50
87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7		6.00
40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5		17.74
75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5		2.35
86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5		3.50
36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8	21 Lakh	17.50
64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5		5.20
65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5		1.95

Figure 2: Train data set part 2

Appendix: Summary: Figure 21.

2.1 Preprocessing

What we have done with the variables:

- **X**: It is useless so we delete it. **Deleted**
- **Name**→**Brand** : Indicates the brand and the model of the car, but we are only interested in the brand because leaving the model may lead to an overfitted model. **Categorical**

- **Location:** As it does not have any NA values and seems correct we do not apply any preprocessing. We know that variable limited us in terms of prediction because we can only obtain results strictly of those locations appearing in the database. But we need to include it because we have few variables and otherwise our model will get worse. We have thought about grouping the countries into every continent in order to generalize, but we have realized we only have data from Asia so it cannot be done. So we finally leave it as it was. **Categorical**
- **Year** → **Age** : Explains the year of fabrication of the car so we replace it with the age of the car because we think that is more representative. **Discrete**
- **Kilometers driven:** It does not have NA values, but it seems to have some outliers, we delete them in the visualization part. **Continuous**
- **Fuel Type:** Initially we had 5 levels: CNG(Compressed Natural Gas), Diesel, Electric, LPG (Liquefied Petroleum Gas) and Petrol but we have deleted the NAs values in Power and other variables (there were few) and the rows removed were those whose fuel value was different from Diesel and Petrol. Finally, we only have two levels: Diesel and Petrol so in total 858 were deleted. **Categorical**
- **Transmission:** Two types: Automatic and Manual, it seems they have correct values. **Categorical**
- **Owner Type:** 4 levels: First, Second, Third and Fourth & above, no NAs. **Categorical**
- **Mileage:** We have seen that there are two units: kmpl and km/kg. Nevertheless, we can not ensure a good conversion because of the different density of the fuel, so we decided to remove them because there were only 46 rows with km/kg value. To be able to work with this variable we should convert it to a numerical variable so we removed the units and now we have a float column. We see there are 0 values (68) that we decided to remove because 0 mileage does not make sense. **Continuous**
- **Engine:** As done before, we delete the units. There are 36 NA values which we decided to delete because they are few. **Continuous**
- **Power:** Almost equal to engine but in this case we have 143 NA value. 56 of them are from CNG fuel, 10 from LPG and 2 from Electric. And by removing those NAs values we also removed the NAs of other columns because they were in the same row. So, in fact, we deleted rows with no information (lots of NAs). **Continuous**
- **Seats:** We had NA values but when we deleted the Power NA values they were also removed. One 0 value is removed as it does not make sense. **Categorical**

- **New Price:** It has 5195 null values, which is the 86% of the total rows, so we decided to eliminate the variable. **(Deleted)**
- **Price:** To feel more comfortable we will convert the Indian currency into euros in order to give a better interpretation of the data. We have use the current exchange price (1 Lakh equals to 1214,2 Euros) **Continuous**

At the beginning of the preprocessing we had 6019 rows and 14 columns, now we have 5779 rows and 12 columns, more or less we have removed 4% of the data. Nothing was deleted in vane, as they would complicate the predicting task instead of helping.

2.2 Data visualization

Now we are going to visualize the data in order to get an approximation of the distribution of the variables and detect some outliers.

BRAND: As there are some brands that only have a few cars, we have deleted them because at the moment of partitioning, if by any chance all of the cars from that brand belong to the test group, our model would not work.

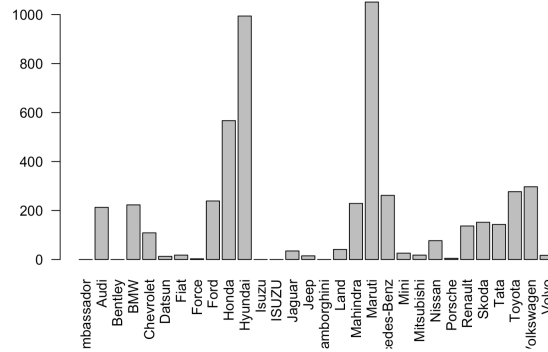


Figure 3: Brand Plot

LOCATION: We can see we have almost the same number of cars of different locations, so it is not a problem.

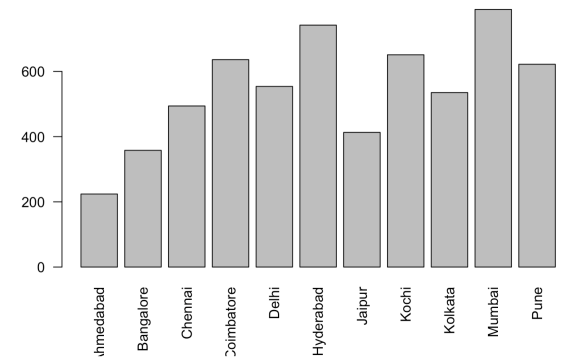


Figure 4: Location plot

AGE: We can see there is not any problem with this variable, apart from a few outliers.

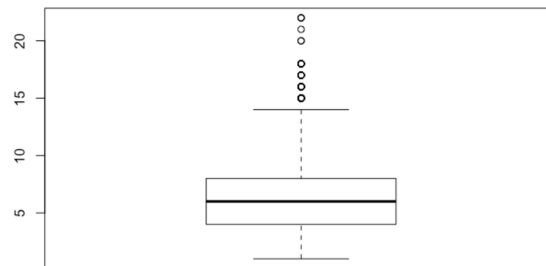


Figure 5: Age boxplot

KILOMETERS DRIVEN: As a first approximation we notice there are some outliers, and we have conclude that is best to remove them.

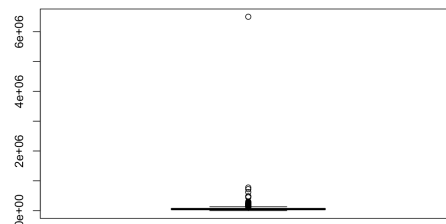


Figure 6: Kilometers driven boxplot

By removing the outliers we improve the data and we can see gaussianity so

let's leave it like that.

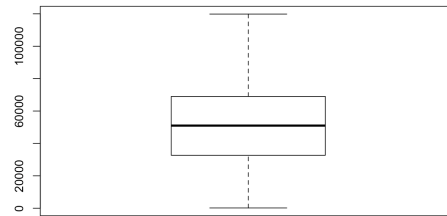


Figure 7: Kilometers driven boxplot, outlier removed

FUEL TYPE: We can see there is not any problem with this variable, as there are almost the same Petrol and Diesel.

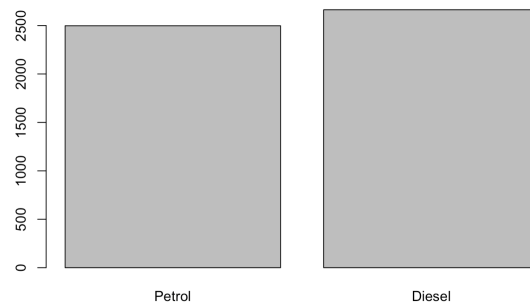


Figure 8: Fuel plot

TRANSMISSION: Although we see that there are much more manual than automatic, it is not a problem because, in general, manual cars are more common than the automatic ones.

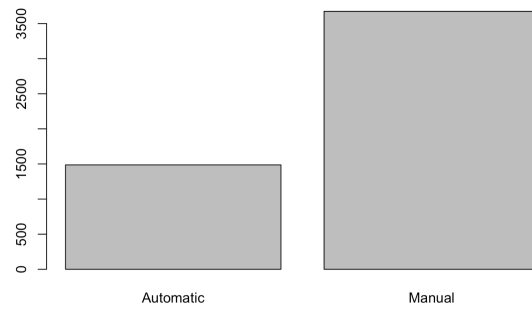


Figure 9: Transmission plot

OWNER TYPE : There are much more cars of first owner type, but it is also the most usual type in real life, so that's not a problem.

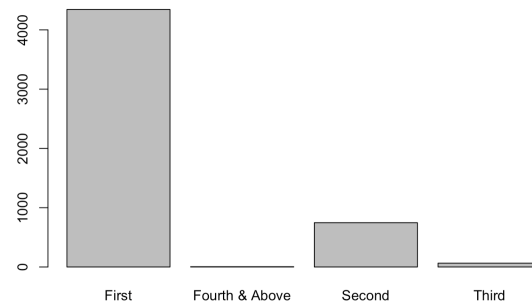


Figure 10: Owner plot

MILEAGE: We can see there is not any problem with this variable.

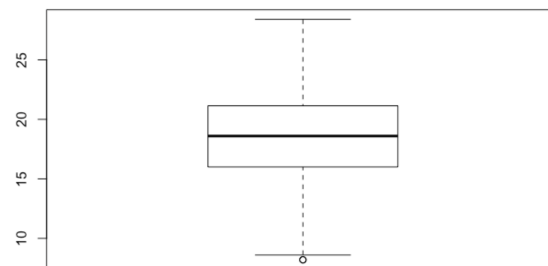


Figure 11: Mileage boxplot

ENGINE: It seems that the data looks like a gaussian but we have many outliers that might affect to the model.

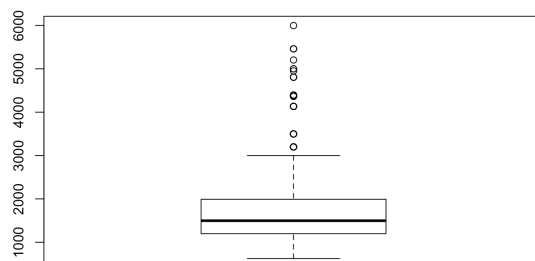


Figure 12: Engine boxplot

Removing the outliers we obtained a good result.

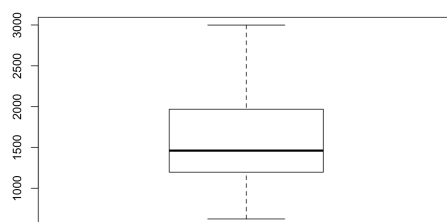


Figure 13: Engine boxplot without outliers

POWER: It happens the same as in Engine, we need to remove the outliers and also apply a log transformation in order to make it look more gaussian.

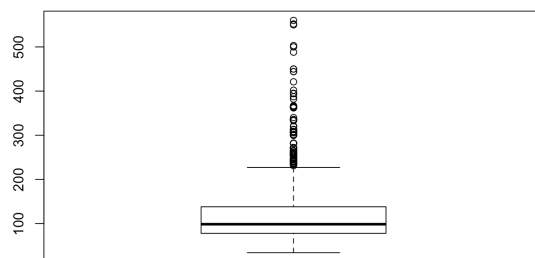


Figure 14: Power boxplot

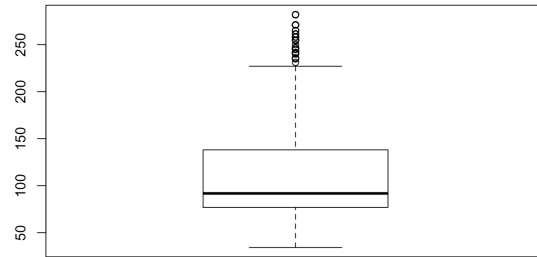


Figure 15: Power boxplot without outliers

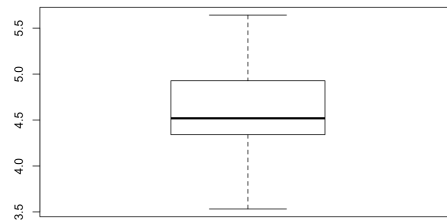


Figure 16: Power boxplot without outliers log transformation

PRICE:

This variable does not seem gaussian, or maybe it is gaussian distributed but with an other link function different from the identity.

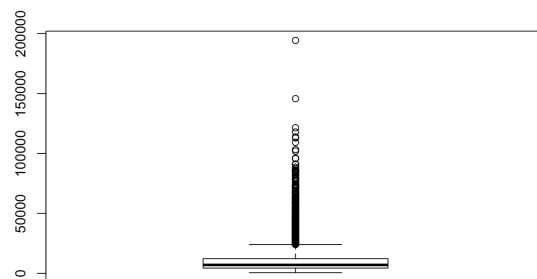


Figure 17: Price boxplot

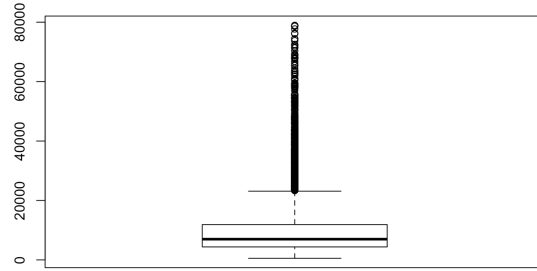


Figure 18: Price boxplot without outliers

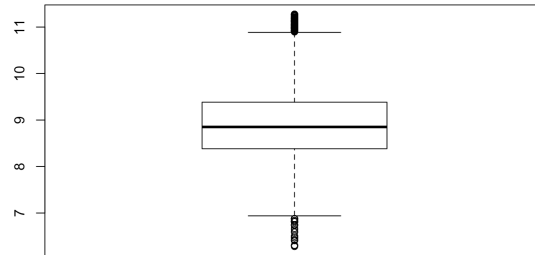


Figure 19: Price boxplot without outliers log transformation

We do a boxcox plot so we can know which transformation has to be applied. We saw that the optimal lambda is approximately 0 so a log transformation would be the best choice. That means that in the GLM model we will use a log link function.

Even though the gaussianity is usually wanted for the explanatory variables, after doing some comparisons, we have come to the conclusions that it is best for the models if we try to reach gaussianity also in the target variable.

By removing the outliers of the variables we have done a reduction of the data and we finally have 5161 entries to analyze.

3 Resampling

Once the preprocessing has been done and analyzed the data behaviour, we move on to the resampling section.

First of all, we choose five candidate models: GLM, Lasso regression, Ridge regression, MLP neural network and Random Forest.

3.1 GLM, Lasso Regression and Ridge Regression

As a linear model is very simple and easy to implement, we thought that they were good candidates. Also, they are easy to understand and see which variables are the ones that have more influence. As an alternative to the GLM, we have implemented the LASSO and Ridge regression because in some cases we need to regularize the variables in order to delete those that are insignificant.

GLM: We created a GLM with the logarithm link function because, as we said before, if we apply that function to the target variable we can assume Gaussianity. Then, we used the step command to delete irrelevant variables looking at the AIC coefficient.

LASSO: To obtain the hyper parameter lambda we did a cross-validation and we got a value of 0.0007237471. It means no regularization is required, all the variables are needed.

Appendix: Lambda Cross-Validation Figure [22](#).

RIDGE: As we had done in the LASSO, we looked for the optimal value of the lambda. By using cross-validation as before, we obtained a value of 0.06442905.

Appendix: Lambda Cross-Validation Figure [23](#).

3.2 Neuronal Network: MLP

As we have not obtained good result with linear methods we look for more complex models, the non-linear. The most used non-linear method is the MLP, but as we have a very high capacity of adaptability we have to look for the optimal decay and number of neurons in order to avoid overfitting. We have used a large number of neurons (28) and then regularized the decay by using cv. We obtained a value of 0.03981072.

Appendix: Decay Cross-validation Figure [24](#).

3.3 Random Forest

One of the methods we have chosen is Random Forest, because it works with both categorical and numerical explanatory variables. Furthermore, as linear models have not fulfilled our expectations, non linear models perhaps could fit better the data. However, it is well known that these models can sometimes be overfitted, so we will keep an eye on that by using some techniques to avoid it.

In order to chose the proper number of trees and mtry, we have based our regularization on OOB. Then, the best values were: 256 for ntrees and 10 for the mtry.

3.4 Cross Validation

Once we have computed all the models and obtained the optimal hyper parameters it is time to perform the cross-validation to decide which is the best. In that case we will use 10-fold 10 cross-validation.

The results obtained are the following:

	GLM <dbl>	LASSO <dbl>	RIDGE <dbl>	MLP <dbl>	RF <dbl>
RMSE	3269.6971829	3272.51432	3434.6913045	2144.2638190	2136.8410682
Rsquare	0.9223578	0.92252	0.9200032	0.9666787	0.9760793
NRMSE	28.0000000	28.00000	29.4000000	18.3000000	18.3000000
3 rows					

Figure 20: Cross-validation comparison

By taking a look at this picture, we can see that, at first, the best model is the Random Forest, because it is the one that has the least RMSE and the largest RSquared. Nevertheless, it will not be the chosen one.

It is noticeable that all the linear models perform almost the same way, with very few differences, especially in Ridge. However, when computing the non linear models, we see that the Random Forest seems to work better, but it is not very clear. Thus, we test them with the test data and we see that the MLP generalizes better, as the Random Forest tends to be overfitted.

4 Results

Finally, we are going to test the performance of MLP using the test data because it was the best solution as we have seen in the cross-validation.

We obtained the following results:

Mean absolute error of the MLP the regression with test data: 1558.994

Percentage error of the MLP 13,69% of error in every prediction in mean. We think it is a very satisfactory result as in the linear regression we have obtained a 17%

Appendix: Predict vs target Figure 27.

In order to look at the most significant variables, we have look at the coefficients in the linear regression. The brand with the largest coefficient is Porsche, with a coefficient of 0.91, which is expected, as Porsche is an expensive brand. Besides, Mercedes have a 0.62 and Mini a 0.89. Moreover, Land Rover has a 0.84 and Jaguar 0.65.

Speaking of numerical variables, the one that has the most influence is the Age, with a coefficient of -0.34, which is obvious that it is negative, as it is known that the older the car is, the cheaper it is. Besides, the power has a coefficient of 0.34 and a 10-seats car has a coefficient of 0.41.

Something to stand out is that the variable Kilometers Driven, which one might think it would be very important, it turns out to be one of the least important. One of the reasons we have thought about is because we are dealing with cars of poor countries, so maybe they do not care that much about the kilometers.

Appendix: Coeficients [Figure 25](#) and [Figure 26](#).

5 Conclusions

By taking into account all the previous analysis, as far as we are concerned there is some variance in the model we cannot explain because we do not have enough data. Thus, we think we have obtained a good approximation given the circumstances.

As we expected, the non-linear models have been able to fit better the data so we have obtained better results. But there is a limit where any model can have a better perform, that is because we cannot have control of all features that affect the price of a car and although if we want to increase the complexity of the model we would not achieve better results, only an overfitting.

6 Extensions and limitations

One way to improve the model could be collecting more data, but not only more observations, but also more variables that affect notably to the price and they were not provided in the data; so that uncertainty results in a higher error. Some examples of that are: maintenance of the car, market demand and extra features. To implement that, for example, we could make a survey to the new owners and ask them for the status of the car, because in case the car is not well preserved, its price drops. Besides, we could take into account the numbers of "visits" to the mechanics in the last 5 years, and many other ways to monitor the status of the car.

However, there are some limitations, because it is not always possible to collect all that data, and even if it was collected, there are some other unknown factors much more difficult to collect, such as the urgency of the owner to sell the car, etc.

7 Appendix

X	Name	Location	Age	Kilometers_Driven	Fuel_Type	Transmission
Min. : 0	Mahindra XUV500 W8	Mumbai	49	790	Min. : 1.000	Min. : 171 CNG : 56 Automatic:1720
1st Qu.:1504	Maruti Swift VDI	Hyderabad	45	742	1st Qu.: 4.000	1st Qu.: 34000 Diesel :3205 Manual :4299
Median :3009	Honda City 1.5 S MT	Kochi	34	651	Median : 6.000	Median : 53000 Electric: 2
Mean :3009	Maruti Swift Dzire VDI	Coimbatore	34	636	Mean : 6.642	Mean : 58738 LPG : 10
3rd Qu.:4514	Maruti Swift VDI BSIV	Pune	31	622	3rd Qu.: 9.000	3rd Qu.: 73000 Petrol :2746
Max. :6018	Hyundai i10 Sportz	Delhi	30	554	Max. :22.000	Max. :6500000
	(Other)	(Other)		2024		
Owner_Type	Mileage	Engine	Power	Seats	Price	
First :4929	17.0 kmpl : 172	Min. : 72	Min. : 34.2	Min. : 0.000	Min. : 0.440	
Fourth & Above: 9	18.9 kmpl : 172	1st Qu.:1198	1st Qu.: 75.0	1st Qu.: 5.000	1st Qu.: 3.500	
Second : 968	18.6 kmpl : 119	Median :1493	Median : 97.7	Median : 5.000	Median : 5.640	
Third : 113	20.36 kmpl: 88	Mean :1621	Mean :113.3	Mean : 5.279	Mean : 9.479	
	21.1 kmpl : 86	3rd Qu.:1984	3rd Qu.:138.1	3rd Qu.: 5.000	3rd Qu.: 9.950	
	17.8 kmpl : 85	Max. :5998	Max. :560.0	Max. :10.000	Max. :160.000	
	(Other) :5297	NA's :36	NA's :143	NA's :42		

Figure 21: Summary of train data set

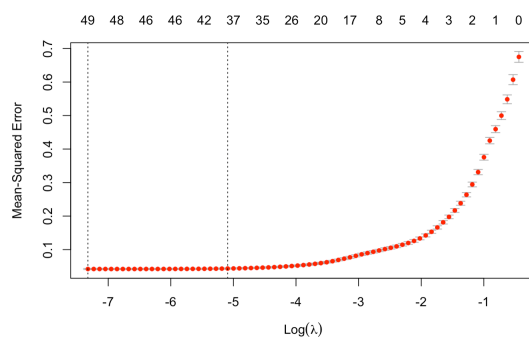


Figure 22: Lambda cross-validation LASSO

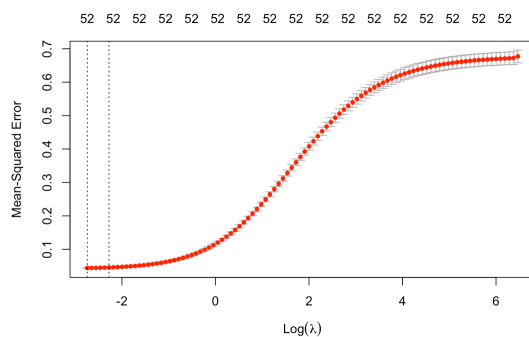


Figure 23: Lambda cross-validation RIDGE


```

Neural Network

3458 samples
11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 3113, 3112, 3112, 3112, 3113, 3112, ...
Resampling results across tuning parameters:

decay      RMSE      Rsquared    MAE
0.01000000 5171.054   0.8057090   3158.077
0.01584893 5182.222   0.8054212   3128.426
0.02511886 5178.266   0.8058994   3169.754
0.03981072 5118.496   0.8098847   3120.021
0.06309573 5125.125   0.8092248   3100.235
0.10000000 5127.540   0.8090651   3125.600
0.15848932 5123.996   0.8089536   3087.672
0.25118864 5200.013   0.8033662   3157.147
0.39810717 5156.195   0.8071520   3113.232
0.63095734 5120.036   0.8101406   3111.282
1.00000000 5187.086   0.8056381   3170.034

Tuning parameter 'size' was held constant at a value of 28
RMSE was used to select the optimal model using the smallest value.
The Final values used for the model were size = 28 and decay = 0.03981072.

```

Figure 24: Decay cross-validation MLP

```

1
(Intercept)      8.915793612
BrandAudi         0.594582060
BrandBentley      .
BrandBMW          0.532941178
BrandChevrolet    -0.307225031
BrandDatsun       -0.253937647
BrandFiat         -0.231892765
BrandForce        0.010849421
BrandFord         -0.058243099
BrandHonda        -0.045810268
BrandHyundai      .
BrandIsuzu        .
BrandISUZU        .
BrandJaguar       0.652830092
BrandJeep         0.121486265
BrandLamborghini .
BrandLand         0.835934526
BrandMahindra     -0.127484582
BrandMaruti       0.054495420
BrandMercedes-Benz 0.617533033
BrandMini         0.891575898
BrandMitsubishi   0.230330895
BrandNissan        -0.003625264
BrandPorsche      0.912540034
BrandRenault      -0.013031443
BrandSkoda        0.031293720
BrandTata         -0.367381271
BrandToyota       0.195185595
BrandVolkswagen   -0.016503430
BrandVolvo        0.314093664
LocationBangalore 0.159647575
LocationChennai   0.035066611
LocationCoimbatore 0.122454784
LocationDelhi     -0.074868881
LocationHyderabad 0.126482445

```

Figure 25: Coefficients Lasso

LocationJaipur	-0.026582474
LocationKochi	-0.008270342
LocationKolkata	-0.245149328
LocationMumbai	-0.061546534
LocationPune	-0.037102288
Age	-0.334928033
Kilometers_Driven	-0.050722450
Fuel_TypeDiesel	0.210475805
TransmissionManual	-0.090985953
Owner_TypeFourth & Above	-0.073266797
Owner_TypeSecond	-0.078191742
Owner_TypeThird	-0.158144531
Mileage	-0.031861883
Engine	0.061179179
Power	0.334960168
Seats4	.
Seats5	-0.098322411
Seats6	-0.110309226
Seats7	.
Seats8	0.013309950
Seats9	0.208286652
Seats10	0.406381418

Figure 26: Coefficients Lasso

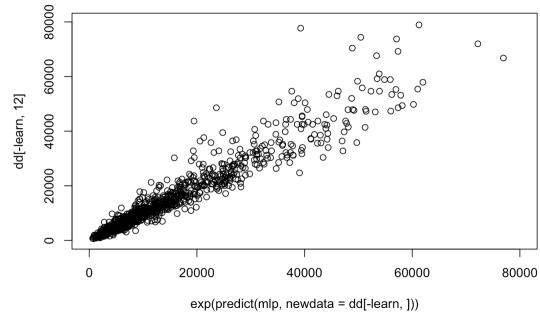


Figure 27: Prediction vs target Final model