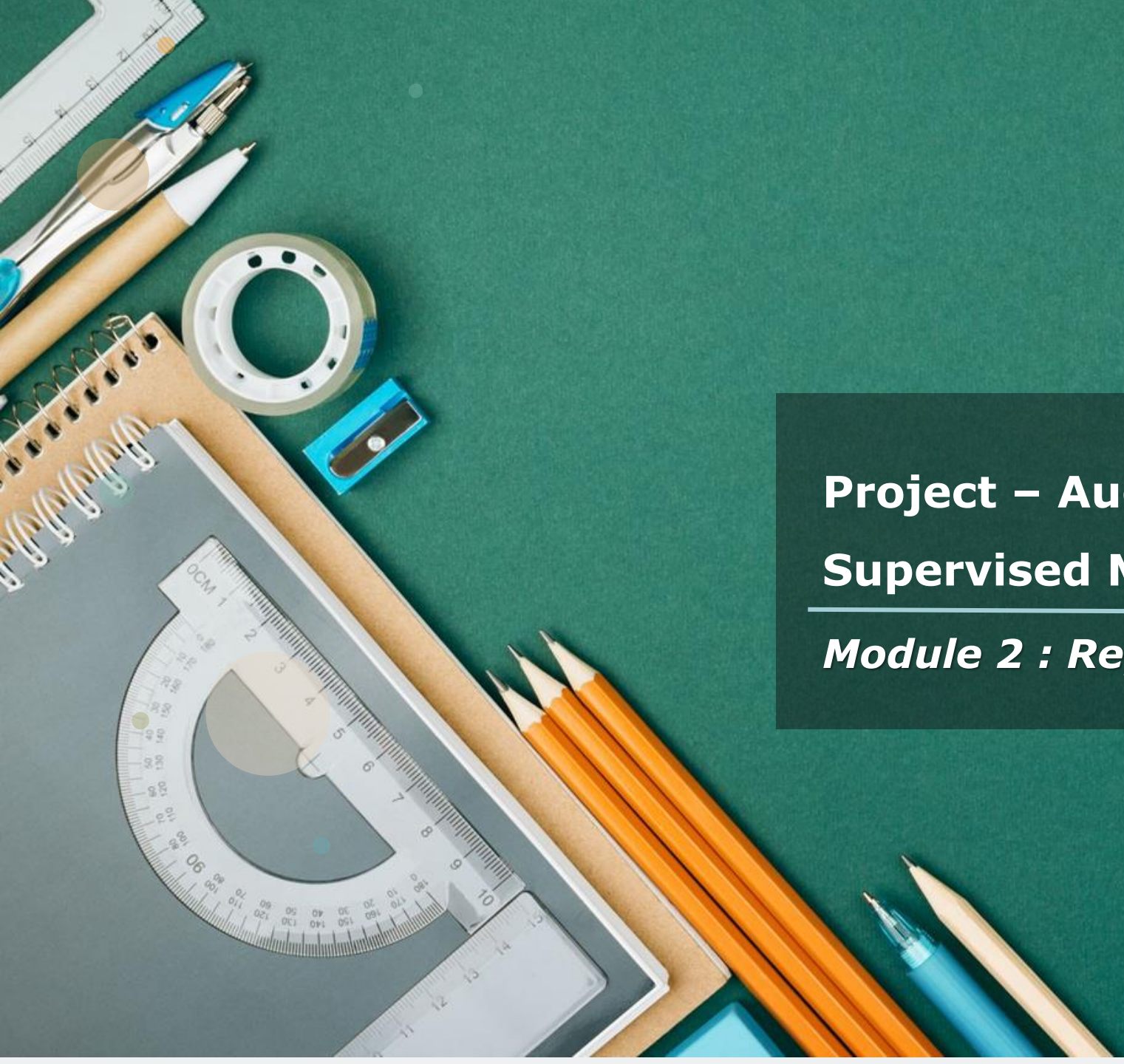


Project – August 2022

Supervised ML

Module 2 : Regression

Main objective

- The main objective of this analysis is to predict price(£) of used Audi cars using a Linear Regression and different regularization regressions.
- This analysis attempts to try both train-test-split and cross-validation to have an overview of how these two methods can lead to different decisions in terms of model selection.
- The data set is split into three sets: training set (60%), validation set (20%), and test set (20%) for cross-validation purpose.

About the data

- The data set used in this analysis is a part of 100,000 UK Used Car Data Set published on Kaggle in July 2020 by a member (Aditya).
- The author scraped the data from 100,000 listings, then cleaned them and removed existing duplicates. The cleaned data were then separated into .csv files corresponding with each car manufacturer.
- The Audi data set was selected for this analysis. This data set has 10,668 records and 9 variables. During the analysis, some duplicates were detected and removed, remaining 10,565 records.

Variable name	Type	Description
model	string	Model of a car
year	integer	Manufacture year
price	integer	Selling price
transmission	string	Transmission type
mileage	integer	Mileage of a car
fuelType	integer	Fuel type
tax	integer	Current tax
mpg	float	Miles per gallon (or equivalent number for electric cars)
engineSize	float	Size of a car engine

Data Exploration

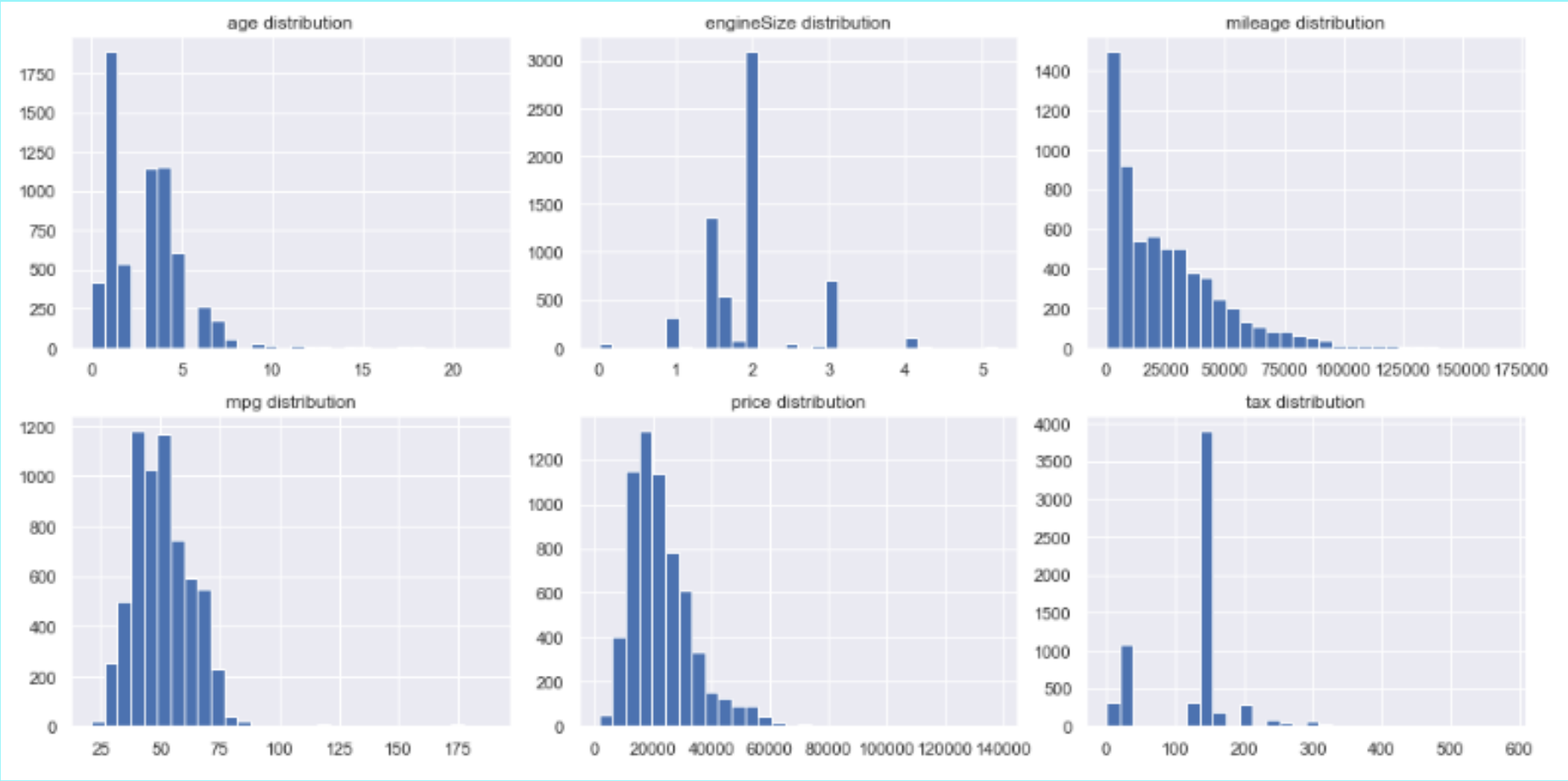
- ❑ After removing duplicates, the EDA is conducted on the training set
- ❑ Select cars that manufactured before 2021 (year <= 2020)
- ❑ Add in a new column, age, and remove year
- ❑ The data info is below :

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6339 entries, 640 to 8728
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   model         6339 non-null   object
 1   year          6339 non-null   int64
 2   price         6339 non-null   int64
 3   transmission  6339 non-null   object
 4   mileage       6339 non-null   int64
 5   fuelType      6339 non-null   object
 6   tax           6339 non-null   int64
 7   mpg           6339 non-null   float64
 8   engineSize    6339 non-null   float64
dtypes: float64(2), int64(4), object(3)
memory usage: 495.2+ KB
```

	year	price	mileage	tax	mpg	engineSize
count	6339.000000	6339.000000	6339.000000	6339.000000	6339.000000	6339.000000
mean	2017.098438	22930.883420	24892.295946	126.260451	50.776037	1.939580
std	2.139155	11547.633121	23276.691439	66.976149	12.933722	0.603108
min	1998.000000	1490.000000	10.000000	0.000000	21.000000	0.000000
25%	2016.000000	15000.000000	6000.000000	125.000000	40.900000	1.500000
50%	2017.000000	20450.000000	19100.000000	145.000000	49.600000	2.000000
75%	2019.000000	27995.000000	36952.500000	145.000000	58.900000	2.000000
max	2020.000000	137995.000000	168017.000000	580.000000	188.300000	5.200000

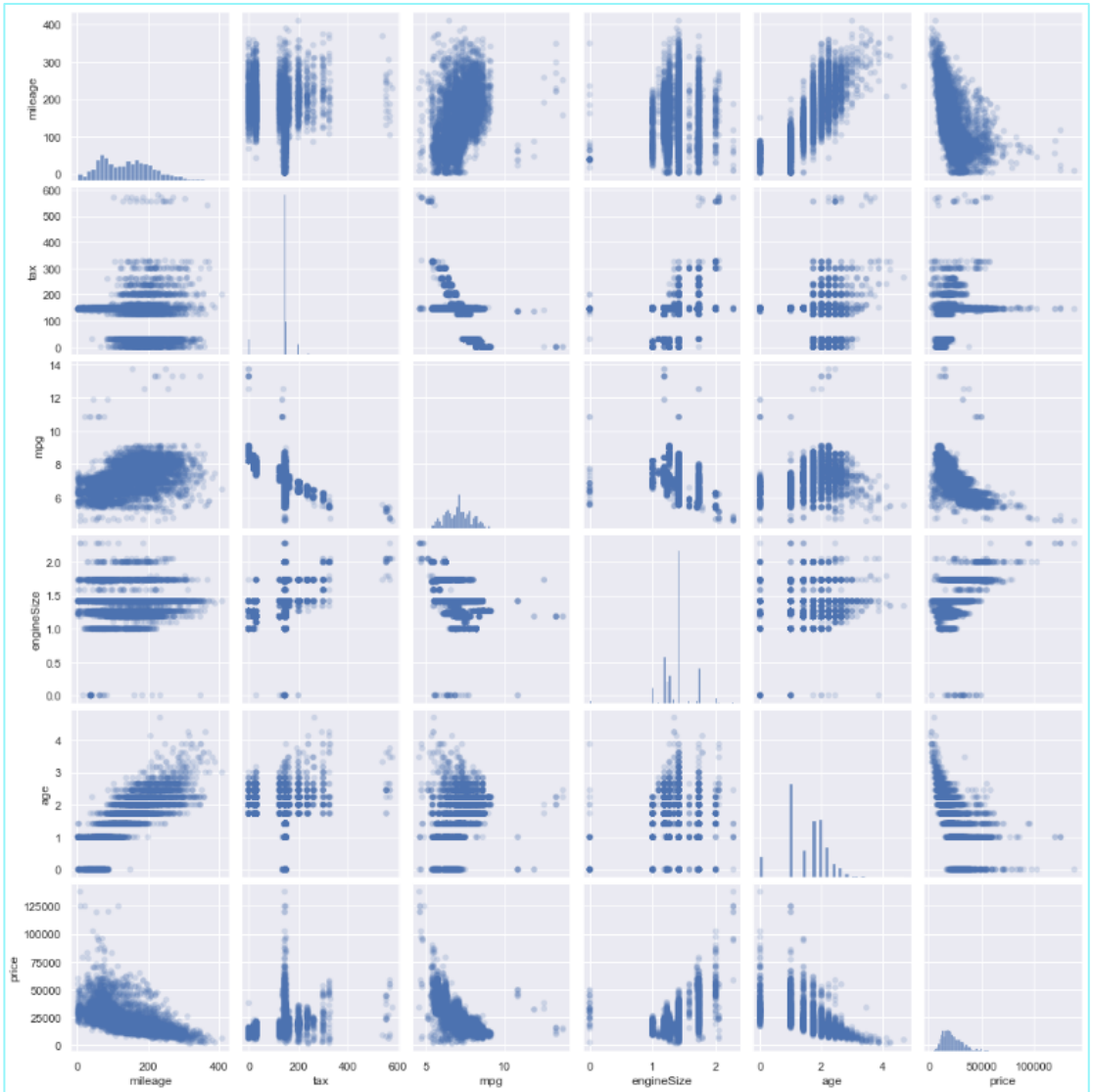
	model	transmission	fuelType
count	6339	6339	6339
unique	24	3	3
top	A3	Manual	Diesel
freq	1137	2601	3369

Data Exploration



- Except for tax and mpg, all features are right-skewed, and there are zero values in engine Size (electric cars).
- Square root transformation might be a good choice to eliminate the skewness in this case

Data Exploration

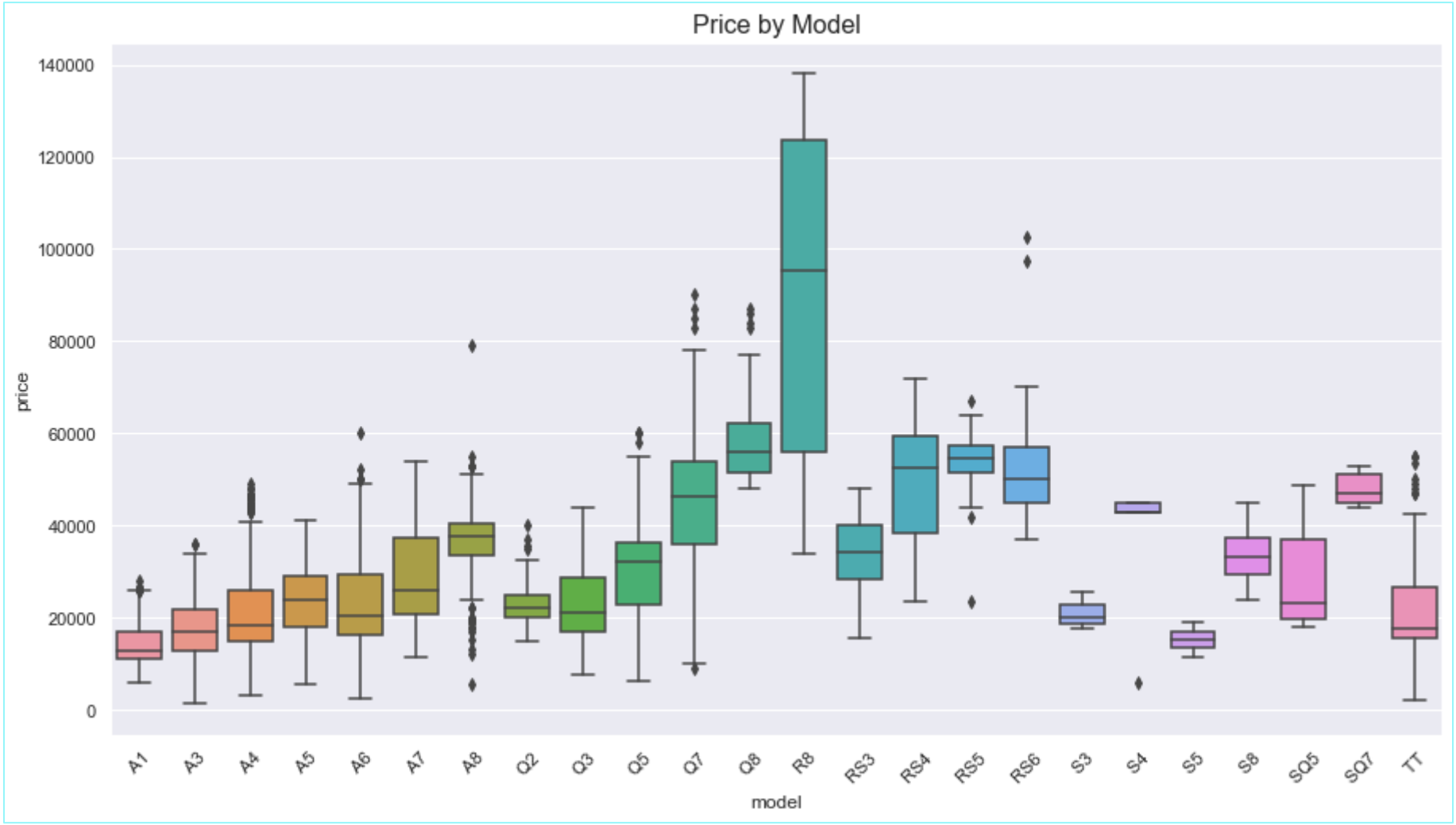


After taking square roots of skewed features, check all numerical variables again. This pair plot shows that:

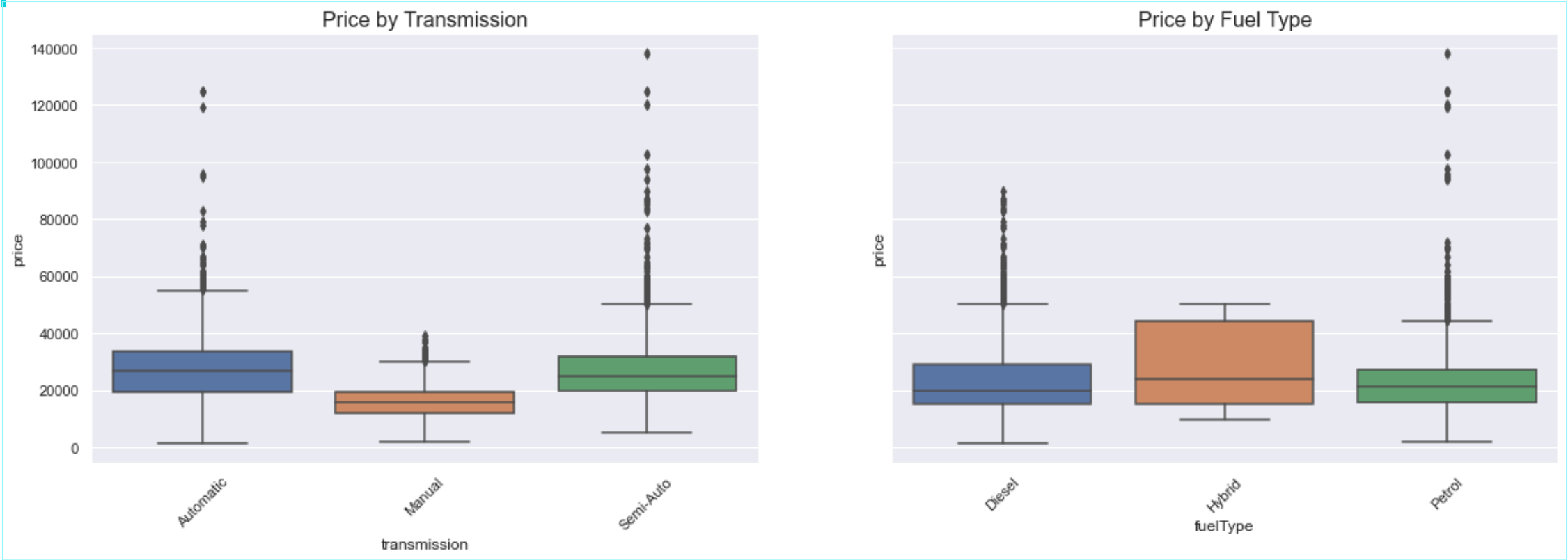
- ❑ price has a linear relationship with mileage. It looks quite like polynomial.
- ❑ age also has a linear relationship with mileage (the older the more miles). This is multicollinearity

Next pages show box plots of categorical features.

Data Exploration



Data Exploration



On average, car prices vary among models, transmission, and fuel types.

Feature engineering

Feature engineering is applied in order to create model variations. Each model is evaluated based on its root mean square error.

- ❑ Apply one-hot encoding to categorical features: model, transmission, and fuel Type
- ❑ Apply square root transformation to features that have a skew value greater than 0.75 (engine Size, mileage, age)
- ❑ Scale numerical features
- ❑ Add polynomial features

All these engineering steps are performed on training and validation sets, using K-fold cross-validation with k=5.

- ❑ The model that has encoded features performs better. The significant difference between rmse of tests and train sets can be a sign of overfitting.
- ❑ The transformation doesn't improve models. We just can see that the rmse of test set diminish slightly for the model with transformed dataset.
- ❑ Scaling features is a preparation for regularization later. RMSEs of both training set and test set should stay the same

	Model	Number of features	RMSE train	RMSE test
0	not encoded	5	5390.123953	5925.685021
1	one hot encoded	32	3797.939697	4077.817927
2	not encoded + squareroot	5	6066.276702	6502.193414
3	one hot encoded + squareroot	32	3856.858717	4041.610554
4	one hot encoded + squareroot + scaled	32	3856.858717	4041.610554

Feature engineering

- We will add polynomial features to the latest model (encoded, square root transformed, and scaled).
- It seems that the third polynomial degree transformation returns the best model.

	Model	Number of features	RMSE train	RMSE test
0	Degree = 1	32	3797.939697	4077.817927
1	Degree = 2	47	2955.533466	3259.088528
2	Degree = 3	82	2702.177641	2943.187492
3	Degree = 4	152	2615.326499	3216.172149
4	Degree = 5	278	2500.43632	6833.84783
5	Degree = 6	488	2351.350655	186008.885556
6	Degree = 7	818	2213.247483	4277978.751895
7	Degree = 8	1313	2085.133766	307876407.791879
8	Degree = 9	2028	1890.932269	257124867143.041931
9	Degree = 10	3029	3024.699429	5545775986326.414062

Cross-validation and Regularization

- Use the same data pipeline: one-hot encoding, square root transformation, standard scaling, and polynomial features adding.
- Use cross-validation to fit the linear regression model again, and then attempt to tune the hyperparameter to find a proper combination of alpha and polynomial degree for regularization. Regularized models include Lasso, Ridge, and Elastic Net.
- Each model is evaluated based on its average root mean squared error (from 5 folds).

- Iterate over different polynomial degree (1, 2, 3) and alphas.
- Result tables are sorted by RMSE in ascending order.

Linear

	Average RMSE
Degree = 3	2938.690720
Degree = 2	3166.363135
Degree = 1	3947.217131
Degree = 4	4183.292984
Degree = 5	28031.217946
Degree = 6	255972.035234

Ridge

	Average RMSE
Degree = 3, alpha = 0.005	2938.866484
Degree = 3, alpha = 0.01	2939.043724
Degree = 3, alpha = 0.05	2940.510965
Degree = 3, alpha = 0.1	2942.452495
Degree = 3, alpha = 0.3	2951.035146

Lasso

	Average RMSE
Degree = 3, alpha = 0.1	2936.286653
Degree = 3, alpha = 0.3	2936.352720
Degree = 3, alpha = 0.05	2936.389179
Degree = 3, alpha = 0.01	2938.137137
Degree = 3, alpha = 0.005	2938.413383

Elastic Net

	Average RMSE
Degree = 3, alpha = 0.005	3300.985636
Degree = 3, alpha = 0.01	3380.697864
Degree = 3, alpha = 0.05	3495.811694
Degree = 2, alpha = 0.005	3514.675316
Degree = 3, alpha = 0.1	3544.232076

- The metrics among Lasso, Ridge, and Linear regression are not significantly different.
- The best model is Lasso Regression with polynomial degree = 3 and alpha = 0.1.
- Elastic Net has the highest RMSE.

Model	Average RMSE	Average R2
Lasso	2936.286653	0.935071
Linear	2938.690720	0.934947
Ridge	2938.866484	0.934938
Elastic Net	3300.985636	0.917290

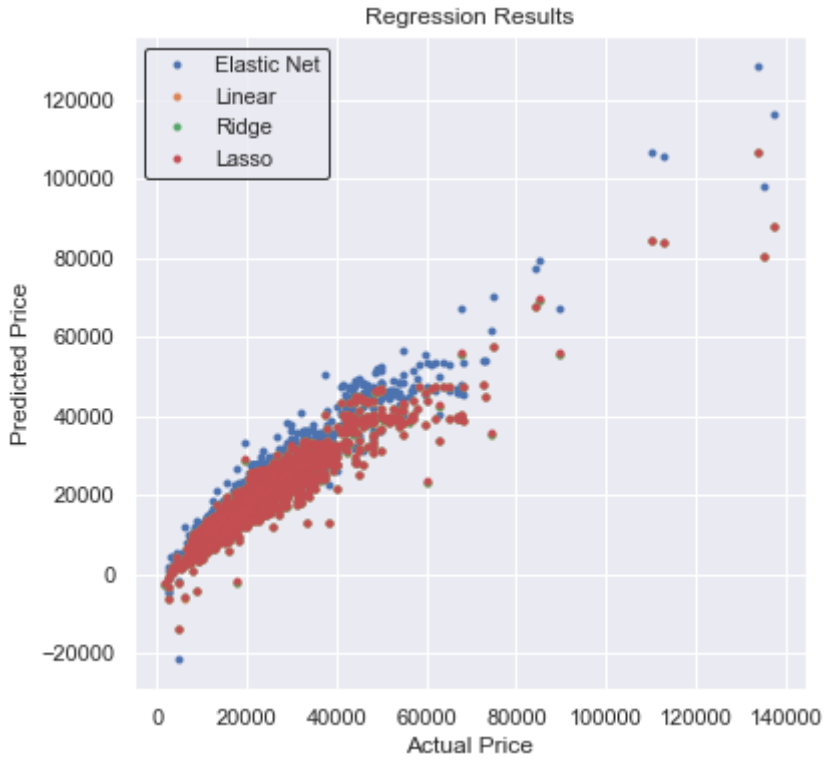
Predict on the test set

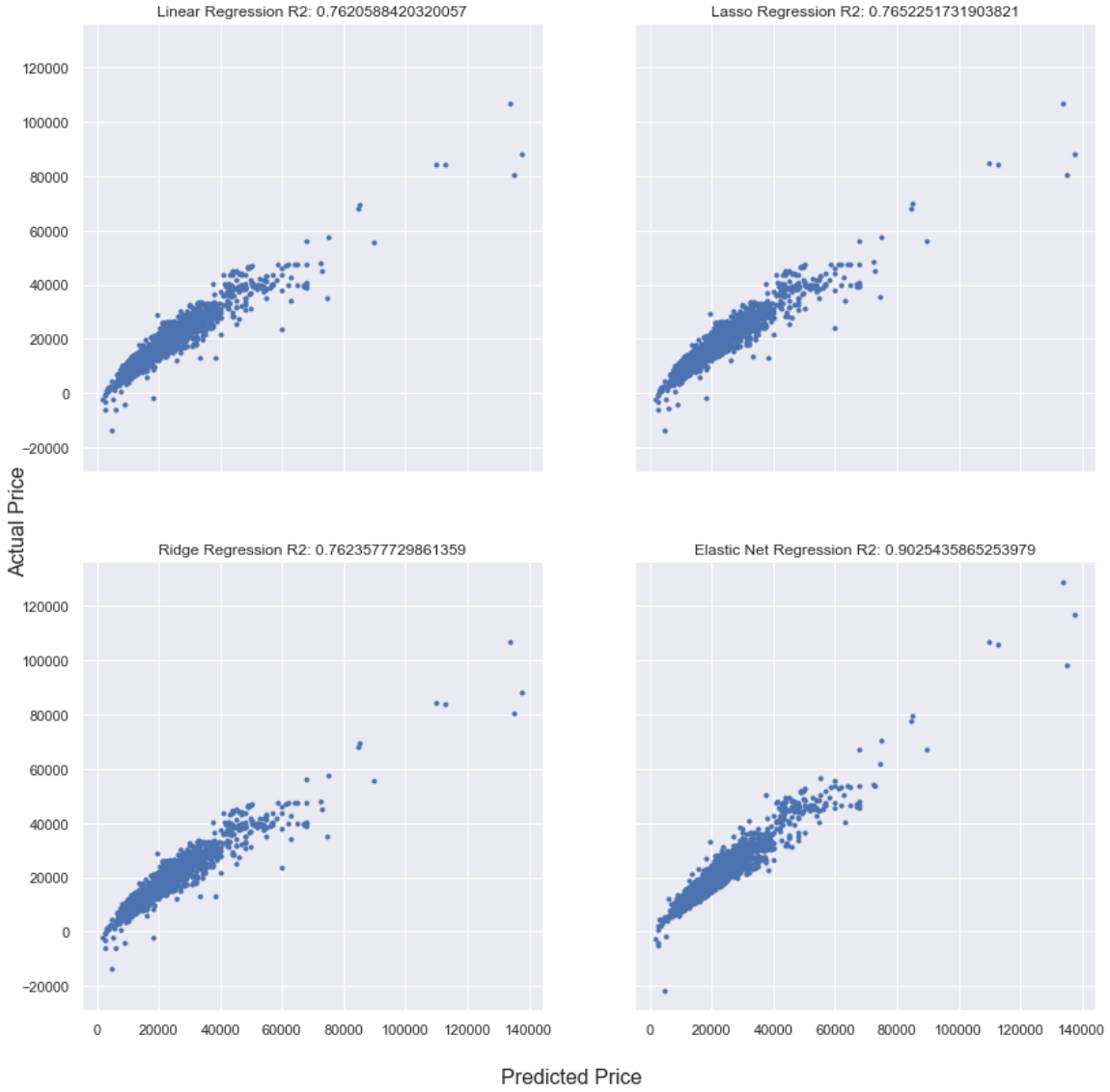
Fit four models to the unseen test set and calculate the R2 score for each model.

- ❑ Linear regression with third degree polynomial features
- ❑ Lasso regression with third degree polynomial features and alpha = 0.1
- ❑ Ridge regression with third degree polynomial features and alpha = 0.005
- ❑ Elastic Net regression with third degree polynomial features and alpha = 0.005

Next page shows scatter plots (true vs predicted price) and R2 scores.

Predict on the test set





Predict on the test set

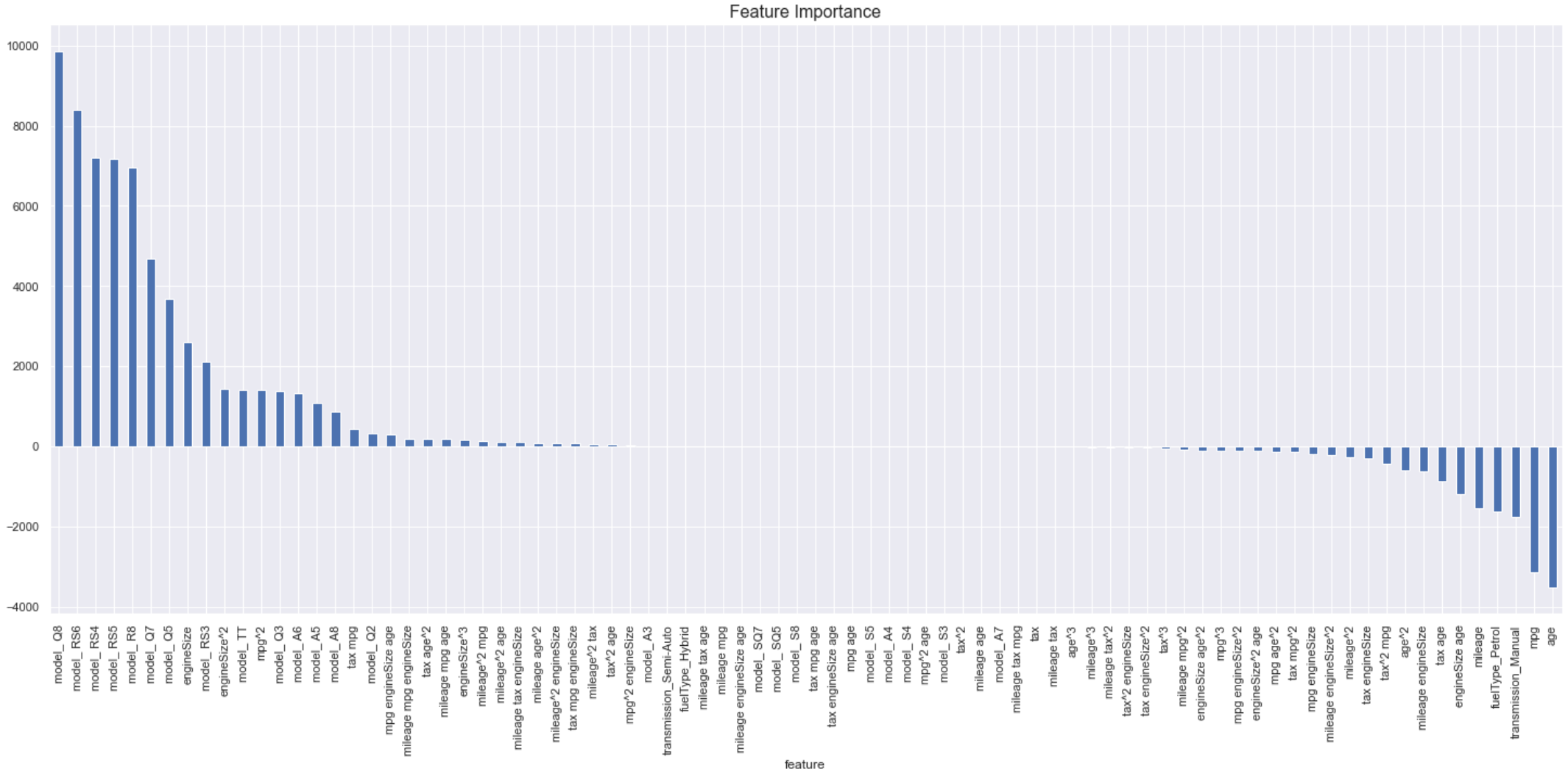
- These plots show that the elastic net regression perform well on the test set ! What is not expected in the train and validation steps. This can be explained by the test data that is concentrated around some particular x-value where the elastic model happens to predict well and see better performance on test than training.

- The other models (lasso, ridge, linear) preform at the rate of 75%.

- So, we will ignore this model, and note that Lasso Regression is the best model (R2 = 0.7652), and it also shrinks some of the coefficients. Looking back at the box plots of price by model and fuel type, we can see that these shrunk features are rare categories in our data set (or their prices do not vary much)

- The main drivers of this model are features that indicate whether the car model is Q8, RS6, RS5, RS4, R8, Q7, or Q5. These are all derived from the categorical feature - model. Among numerical features, age and mpg have the strongest predictive power. Most interaction terms and polynomial features have low estimates in comparison to others.

Predict on the test set



Conclusion

This analysis shows that feature engineering can have a large effect on the model performance, and if the data are sufficiently large, cross-validation should be preferred over train-test-split to construct model evaluation.

In this case, even though the predictors have high multicollinearity, their coefficients were not shrunk by the Lasso model, and it is shown that regularization does not always make big improvement on a given model.

In the end, the Elastic Net regression has the highest $R2$ when predicting on the test set, and categories of car model appear to be the most important features to predict a car price. Also, Lasso did shrink some of the features that are not so important in terms of prediction.

While researching further analysis, we found a suggestion of using grouped Lasso when a model have categorical features, which is worth trying in this case.