Supervised Learning: Regression

IBM Machine Learning - Project 2
Thanh Huynh
March 2021

Main objective

- The main objective of this analysis is to predict price(£) of used Ford 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.

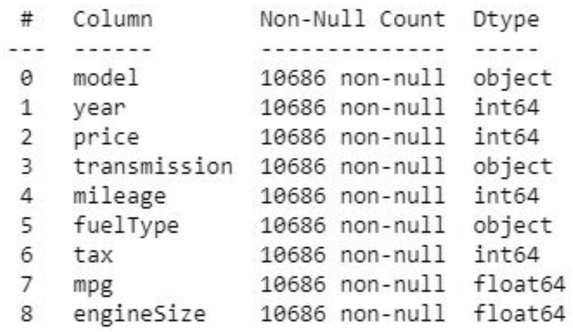
About the data

- 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 Ford data set was selected for this analysis. This data set has 17,965 records and 9 variables. During the analysis, some duplicates were detected and removed, remaining 17,811 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

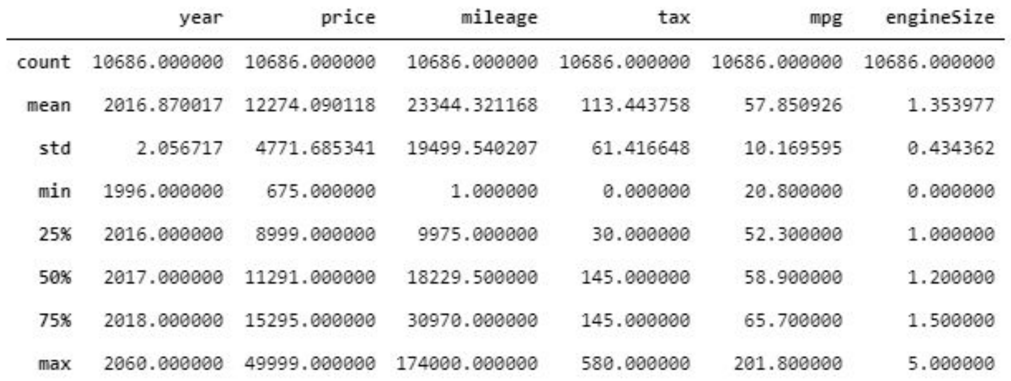
- After removing duplicates, the EDA is conducted on the training set.
- The data description is as follows.



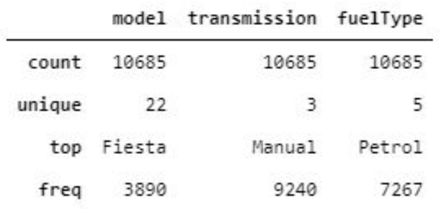
```
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   model         10686 non-null  object 
 1   year          10686 non-null  int64  
 2   price         10686 non-null  int64  
 3   transmission  10686 non-null  object 
 4   mileage       10686 non-null  int64  
 5   fuelType      10686 non-null  object 
 6   tax           10686 non-null  int64  
 7   mpg           10686 non-null  float64
 8   engineSize    10686 non-null  float64
```

Data exploration

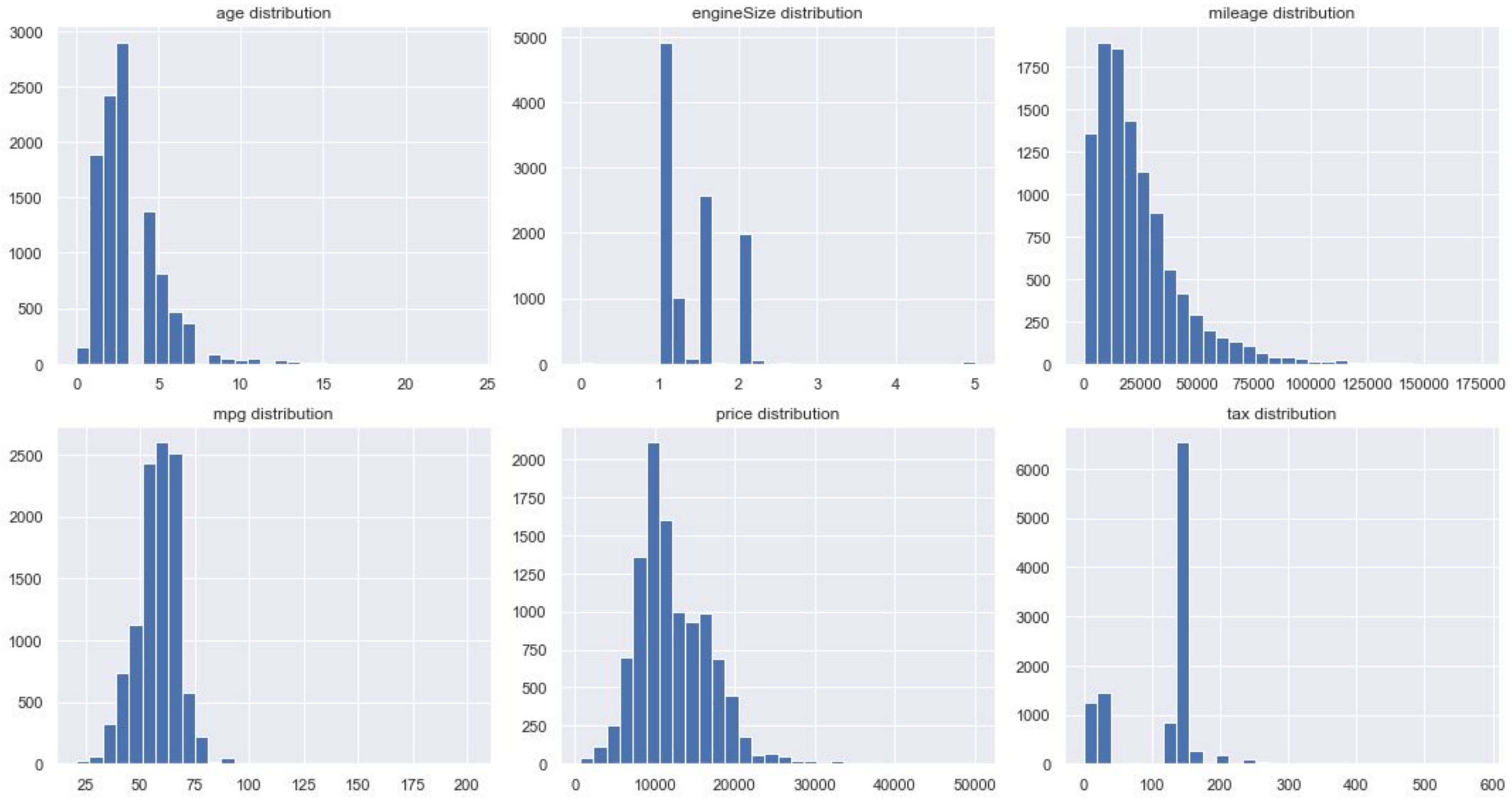
- Select cars that manufactured before 2021 (year <= 2020)
- Add in a new column, age, and remove year.



	year	price	mileage	tax	mpg	engineSize
count	10686.000000	10686.000000	10686.000000	10686.000000	10686.000000	10686.000000
mean	2016.870017	12274.090118	23344.321168	113.443758	57.850926	1.353977
std	2.056717	4771.685341	19499.540207	61.416648	10.169595	0.434362
min	1996.000000	675.000000	1.000000	0.000000	20.800000	0.000000
25%	2016.000000	8999.000000	9975.000000	30.000000	52.300000	1.000000
50%	2017.000000	11291.000000	18229.500000	145.000000	58.900000	1.200000
75%	2018.000000	15295.000000	30970.000000	145.000000	65.700000	1.500000
max	2060.000000	49999.000000	174000.000000	580.000000	201.800000	5.000000



	model	transmission	fuelType
count	10685	10685	10685
unique	22	3	5
top	Fiesta	Manual	Petrol
freq	3890	9240	7267


age distribution
0
500
1000
1500
2000
2500
3000
0
5
10
15
20
25
engineSize distribution
0
1000
2000
3000
4000
5000
0
1
2
3
4
5
mileage distribution
0
250
500
750
1000
1250
1500
1750
0
25000
50000
75000
100000
125000
150000
175000
mpg distribution
0
500
1000
1500
2000
2500
25
50
75
100
125
150
175
200
price distribution
0
250
500
750
1000
1250
1500
1750
2000
0
10000
20000
30000
40000
50000
tax distribution
0
1000
2000
3000
4000
5000
6000
0
100
200
300
400
500
600

Data exploration

- Except for tax and mpg, all features are right-skewed, and also there are zero values in engineSize (electric cars).
- Square root transformation might be a good choice to eliminate the skewness in this case.
- The target is kept the unchanged.

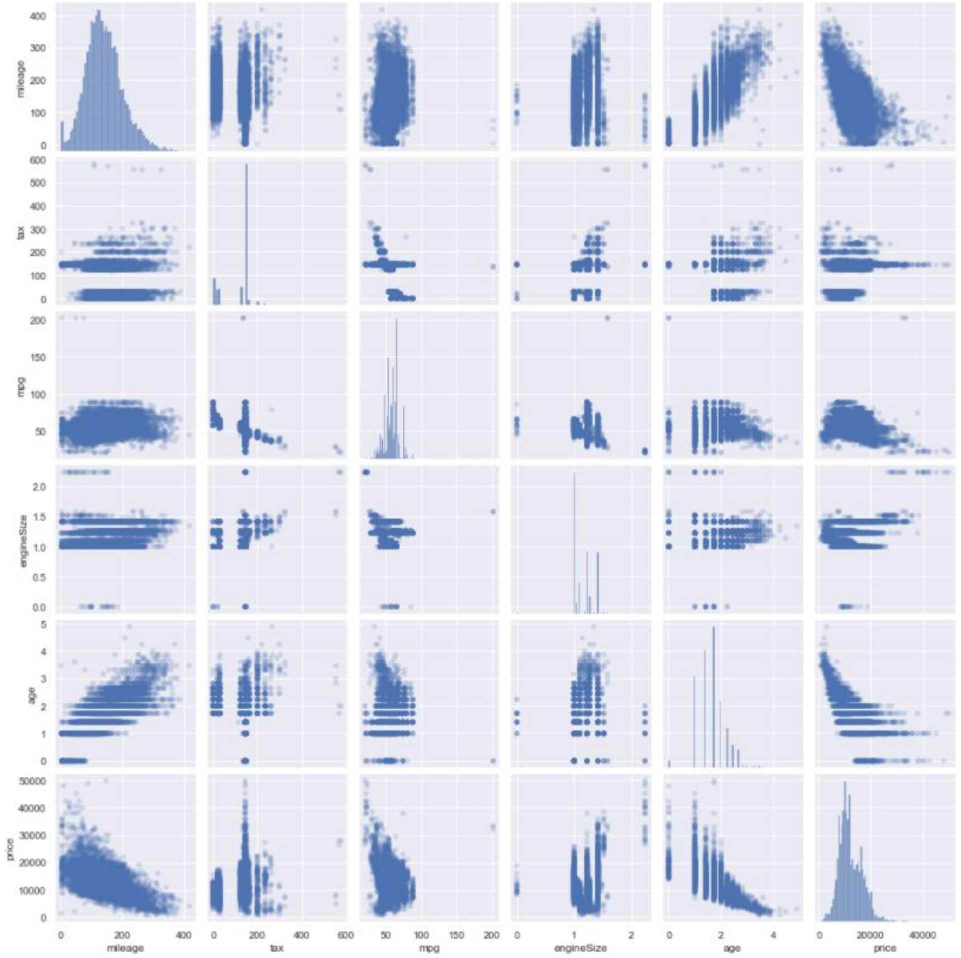
Data exploration

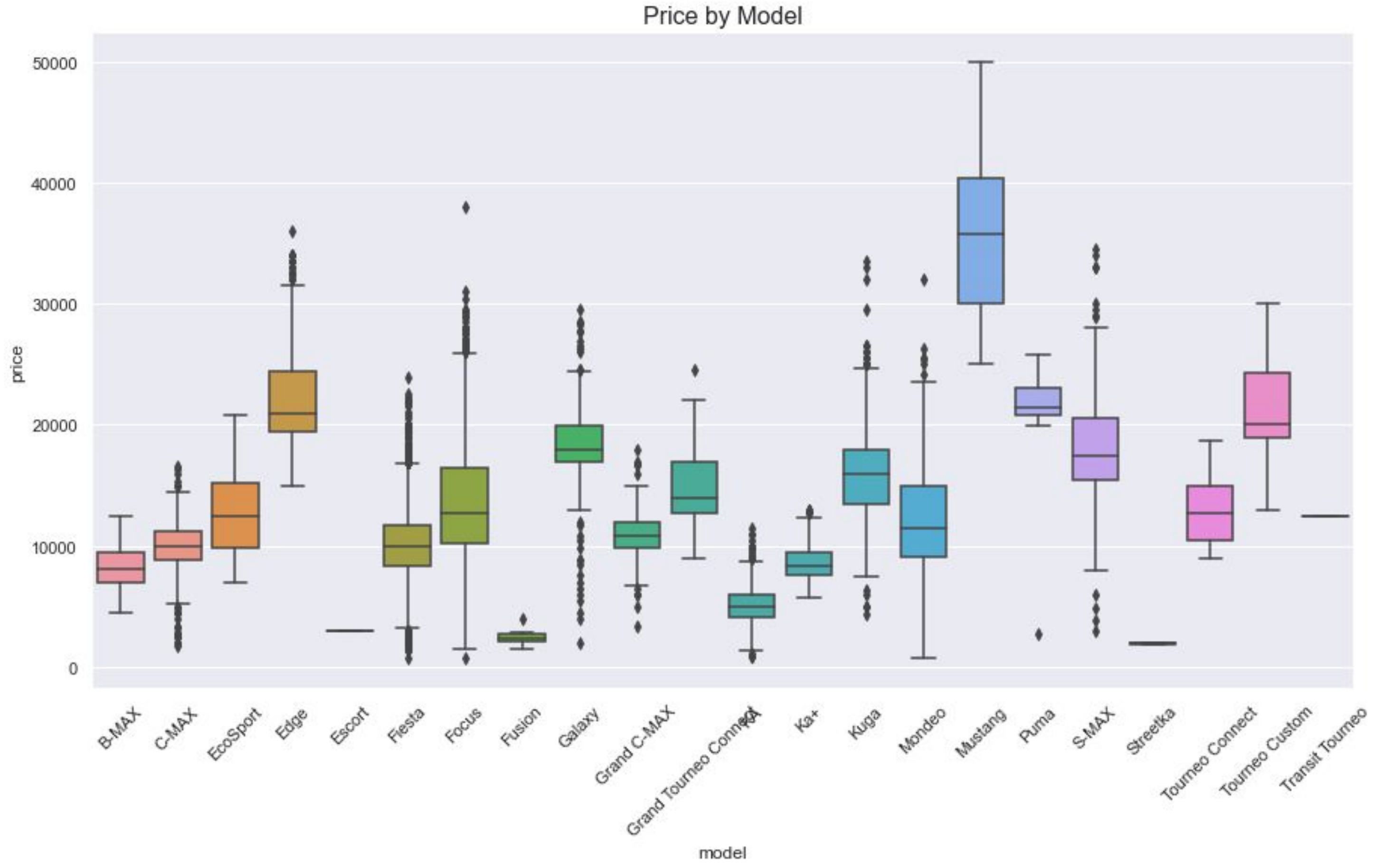
After taking square roots of skewed features, check all numerical variables again. This pair plot shows that:

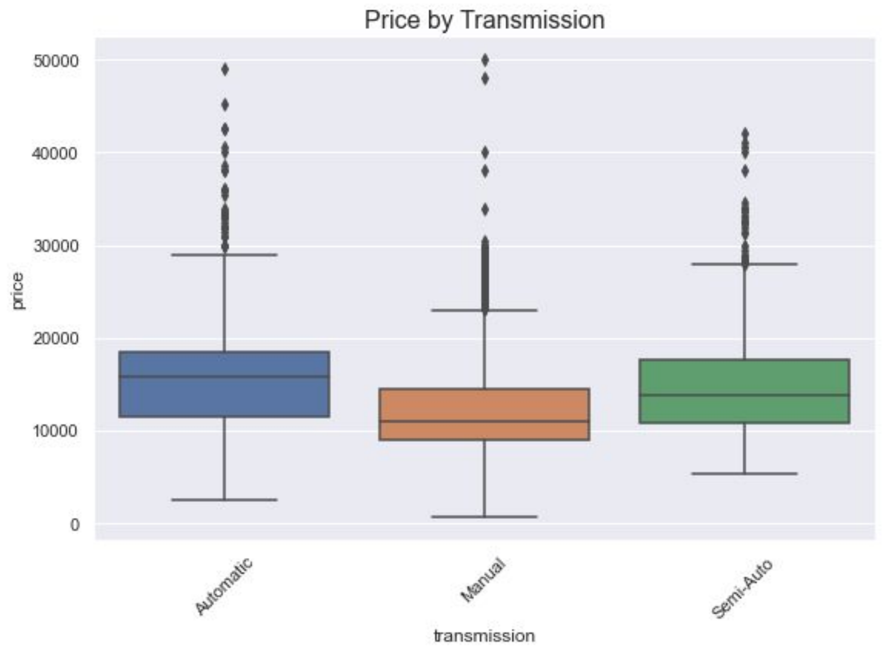
- *age* has a linear relationship with *price*. It looks quite like polynomial.
- *age* also has a linear relationship with *mileage* (the older the more miles). This is multicollinearity.

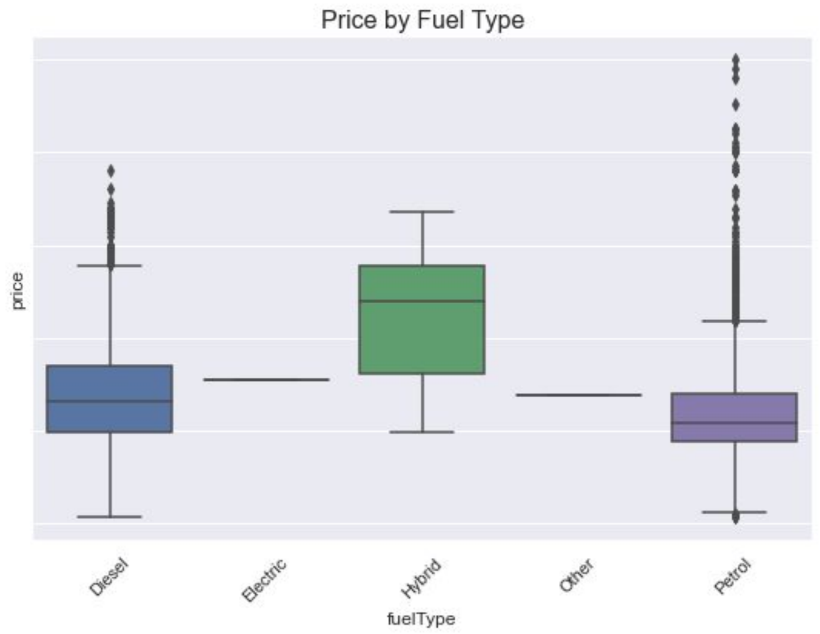
Multicollinearity might be resolved by regularization later.

Next pages show box plots of categorical features.




Price by Model
price
0
10000
20000
30000
40000
50000
model
B-MAX
C-MAX
EcoSport
Edge
Escort
Fiesta
Focus
Fusion
Galaxy
Grand C-MAX
Grand Tourneo Connect
KA
Ka+
Kuga
Mondeo
Mustang
Puma
S-MAX
Streetka
Tourneo Connect
Tourneo Custom
Transit Tourneo





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

Feature engineering and model variations

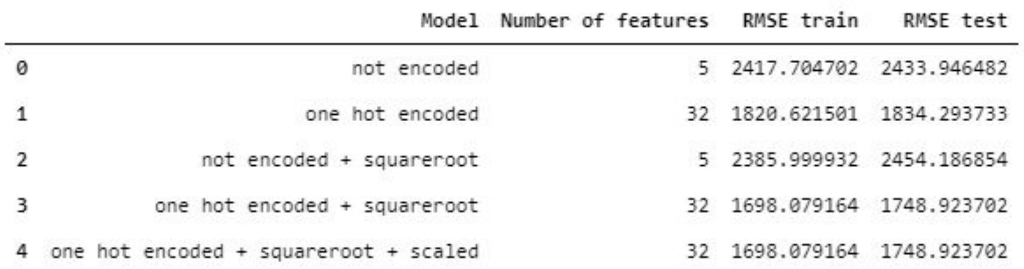
Feature engineering is applied in order to create model variations. Each model is evaluated based on its root mean square error.

- Apply one-hot encoding to categorical features: *model*, *transmission*, and *fuelType*
- Apply square root transformation to features that have a skew value greater than 0.75 (*engineSize*, *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.

Feature engineering and model variations

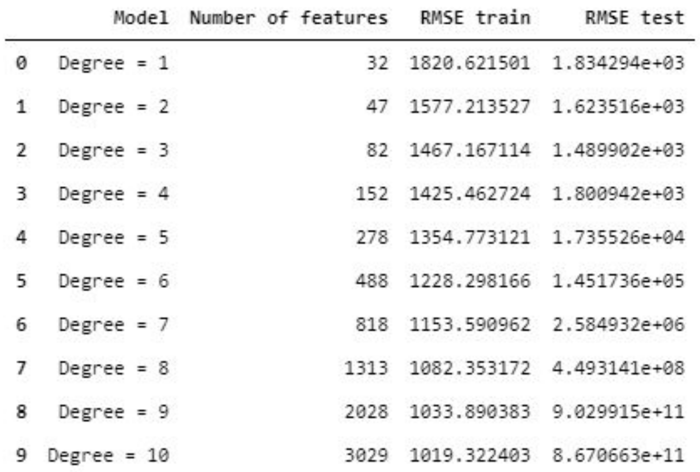
- The model that has encoded features performs better, which is understandable because it has more information to predict the target. RMSEs of validation sets are slightly higher than training sets, which is expected. There is no sign of overfitting.
- The transformation improves all models. The one that has encoded features is the best so far.
- Scaling features is a preparation for regularization later. RMSEs of both training set and validation set should stay the same.



	Model	Number of features	RMSE train	RMSE test
0	not encoded	5	2417.704702	2433.946482
1	one hot encoded	32	1820.621501	1834.293733
2	not encoded + squareroot	5	2385.999932	2454.186854
3	one hot encoded + squareroot	32	1698.079164	1748.923702
4	one hot encoded + squareroot + scaled	32	1698.079164	1748.923702

Feature engineering and model variations

- Add polynomial features to the latest model (encoded, square root transformed, and scaled) and fit the model again.
- It looks like the third polynomial degree transformation returns the best model. At degree 4 and above, as the model gets more and more complex, it starts overfitting.



	Model	Number of features	RMSE train	RMSE test
0	Degree = 1	32	1820.621501	1.834294e+03
1	Degree = 2	47	1577.213527	1.623516e+03
2	Degree = 3	82	1467.167114	1.489902e+03
3	Degree = 4	152	1425.462724	1.800942e+03
4	Degree = 5	278	1354.773121	1.735526e+04
5	Degree = 6	488	1228.298166	1.451736e+05
6	Degree = 7	818	1153.590962	2.584932e+06
7	Degree = 8	1313	1082.353172	4.493141e+08
8	Degree = 9	2028	1033.890383	9.029915e+11
9	Degree = 10	3029	1019.322403	8.670663e+11

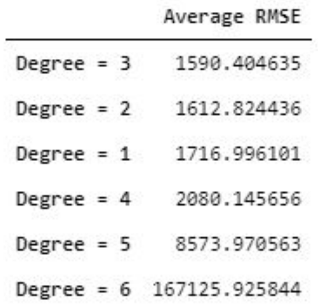
Cross-validation and Regularization

- 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).

Cross-validation and Regularization

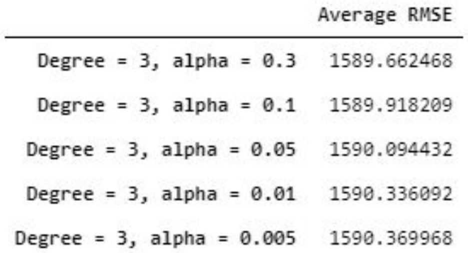
- Iterate over different polynomial degree (1, 2, 3) and alphas.
- Result tables are sorted by RMSE in ascending order.

Linear



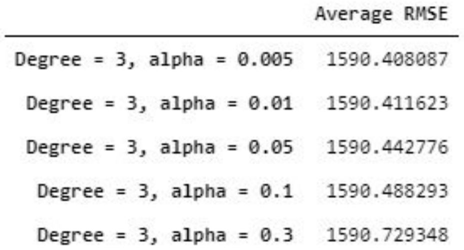
	Average RMSE
Degree = 3	1590.404635
Degree = 2	1612.824436
Degree = 1	1716.996101
Degree = 4	2080.145656
Degree = 5	8573.970563
Degree = 6	167125.925844

Lasso



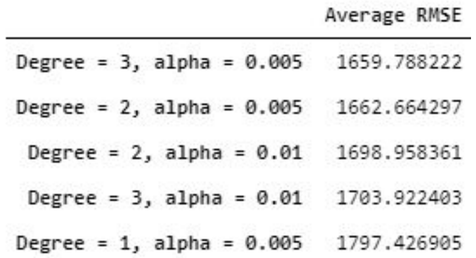
	Average RMSE
Degree = 3, alpha = 0.3	1589.662468
Degree = 3, alpha = 0.1	1589.918209
Degree = 3, alpha = 0.05	1590.094432
Degree = 3, alpha = 0.01	1590.336092
Degree = 3, alpha = 0.005	1590.369968

Ridge



	Average RMSE
Degree = 3, alpha = 0.005	1590.408087
Degree = 3, alpha = 0.01	1590.411623
Degree = 3, alpha = 0.05	1590.442776
Degree = 3, alpha = 0.1	1590.488293
Degree = 3, alpha = 0.3	1590.729348

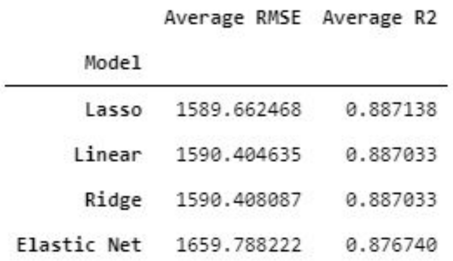
Elastic Net



	Average RMSE
Degree = 3, alpha = 0.005	1659.788222
Degree = 2, alpha = 0.005	1662.664297
Degree = 2, alpha = 0.01	1698.958361
Degree = 3, alpha = 0.01	1703.922403
Degree = 1, alpha = 0.005	1797.426905

Cross-validation and Regularization

- The metrics among Lasso, Ridge, and Linear regression are not significantly different.
- The best model is Ridge Regression with polynomial degree = 3 and alpha = 0.005.
- Elastic Net has the highest RMSE.



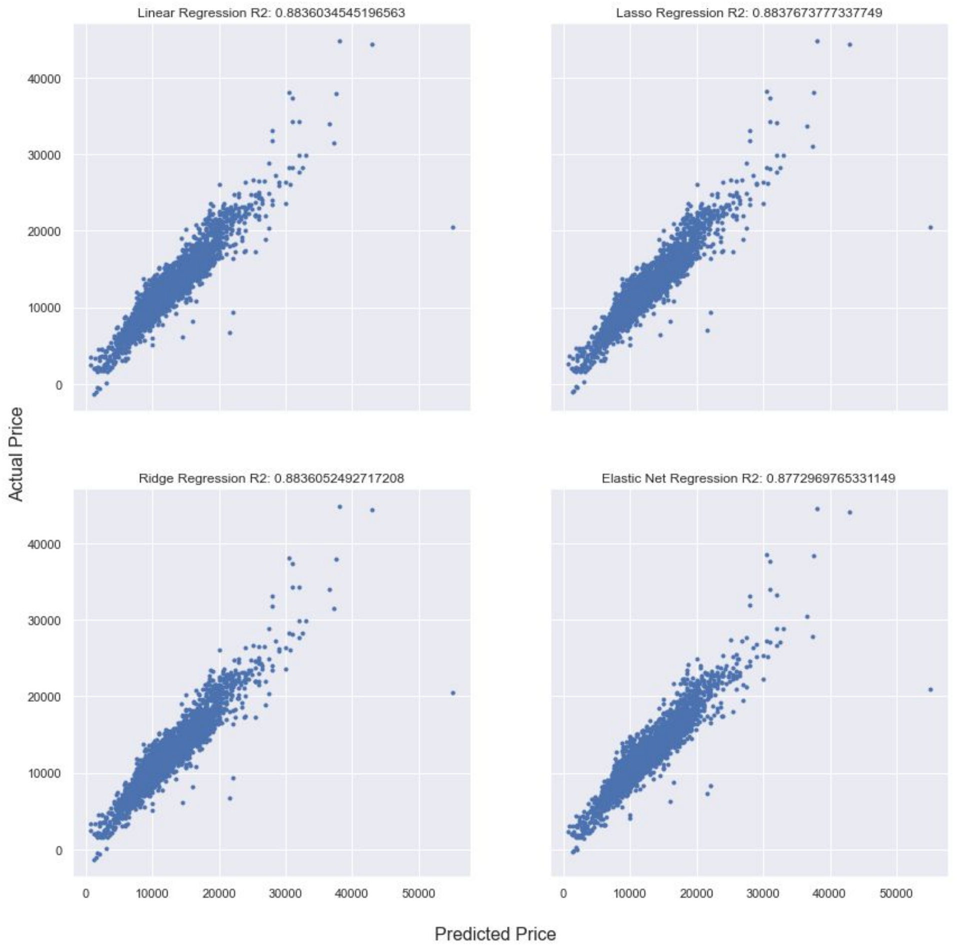
	Average RMSE	Average R2
Model		
Lasso	1589.662468	0.887138
Linear	1590.404635	0.887033
Ridge	1590.408087	0.887033
Elastic Net	1659.788222	0.876740

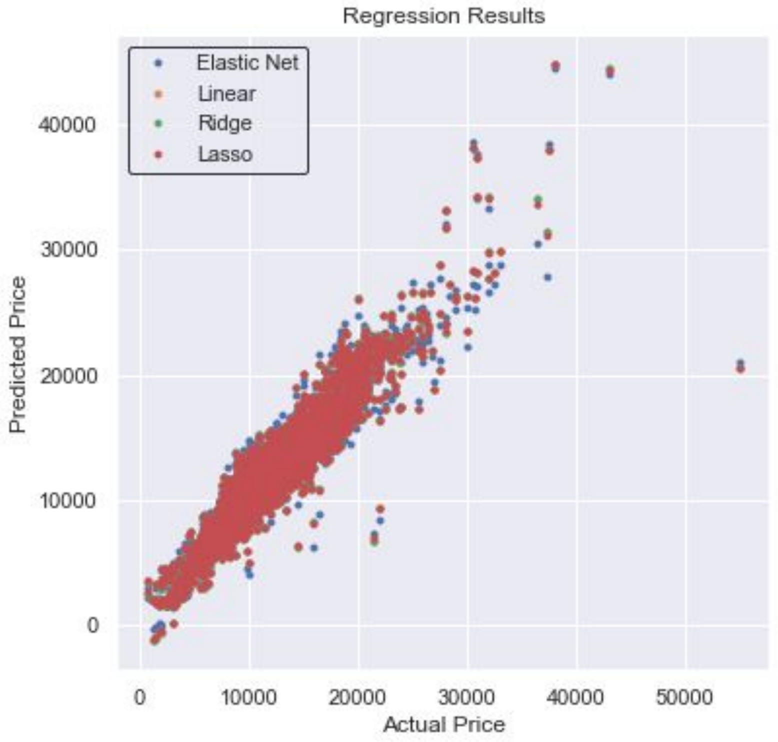
Predict on the test set

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 = 0.3
- Ridge regression with third degree polynomial features and alpha = 0.005
- Elastic Net regression with third degree polynomial features and alpha = 0.005

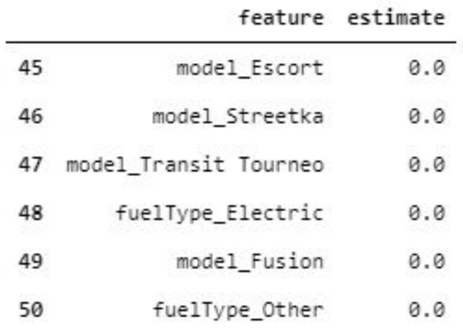
Next page shows scatter plots (true vs predicted price) and R2 scores.


Linear Regression R2: 0.8836034545196563
Lasso Regression R2: 0.8837673777337749
Ridge Regression R2: 0.8836052492717208
Elastic Net Regression R2: 0.8772969765331149
Actual Price
Predicted Price
0
10000
20000
30000
40000
50000


Regression Results
● Elastic Net
● Linear
● Ridge
● Lasso
Predicted Price
Actual Price
0
10000
20000
30000
40000
50000

Predict on the test set

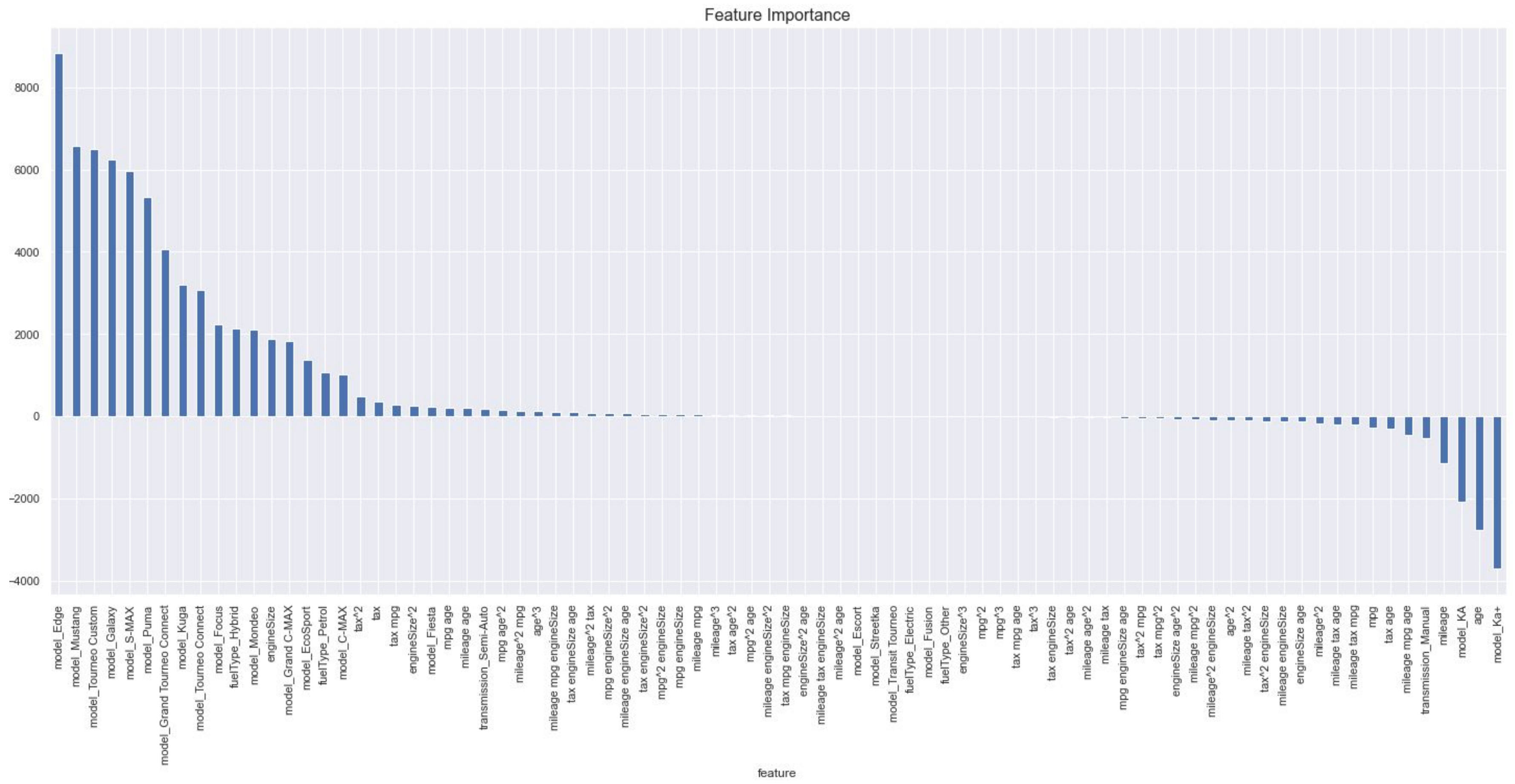
- These plots show that all four models perform pretty well with no sign of overfitting. R2 scores among Lasso, Ridge, and Linear regression are not significantly different.
- Lasso Regression is the best model (R2=0.8837), and it also shrinks some of the coefficients. Looking back at the box plots of 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)



	feature	estimate
45	model_Escort	0.0
46	model_Streetka	0.0
47	model_Transit Tourneo	0.0
48	fuelType_Electric	0.0
49	model_Fusion	0.0
50	fuelType_Other	0.0

Predict on the test set

Next page shows a plot of feature importance of the Lasso Regression. The main drivers of this model are features that indicate whether or not the car model is Edge, Mustang, Tourneo Custom, Galaxy, S-MAX, Puma, or Grand Tourneo Connect. These are all derived from the categorical feature - model. Among numerical features, age and mileage have the strongest predictive power. Most interaction terms and polynomial features have low estimates in comparison to others.


Feature Importance
8000
6000
4000
2000
0
-2000
-4000
model_Edge
model_Mustang
model_Tourneo Custom
model_Galaxy
model_S-MAX
model_Puma
model_Grand Tourneo Connect
model_Kuga
model_Tourneo Connect
model_Focus
fuelType_Hybrid
model_Mondeo
engineSize
model_Grand C-MAX
model_EcoSport
fuelType_Petrol
model_C-MAX
tax^2
tax
tax mpg
engineSize^2
model_Fiesta
mpg age
mileage age
transmission_Semi-Auto
mpg age^2
mileage^2 mpg
age^3
mileage mpg engineSize
tax engineSize age
mileage^2 tax
mpg engineSize^2
mileage engineSize age
tax engineSize^2
mpg^2 engineSize
mpg engineSize
mileage mpg
mileage^3
tax age^2
mpg^2 age
mileage engineSize^2
tax mpg engineSize
engineSize^2 age
mileage tax engineSize
mileage^2 age
model_Escort
model_Streetka
model_Transit Tourneo
fuelType_Electric
model_Fusion
fuelType_Other
engineSize^3
mpg^2
mpg^3
tax mpg age
tax^3
tax engineSize
tax^2 age
mileage age^2
mileage tax
mpg engineSize age
tax^2 mpg
tax mpg^2
engineSize age^2
mileage mpg^2
mileage^2 engineSize
age^2
mileage tax^2
tax^2 engineSize
mileage engineSize
engineSize age
mileage^2
mileage tax age
mileage tax mpg
mpg
tax age
mileage mpg age
transmission_Manual
mileage
model_KA
age
model_Ka+
feature

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 my 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 Lasso 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, I found a suggestion of using grouped Lasso when a model have categorical features, which is worth trying in this case.

Jupyter Notebook can be found here:
https://github.com/thuynh323/IBM-Machine-Learning/blob/master/2-Supervised-Learning-Regression/Project-2.ipynb