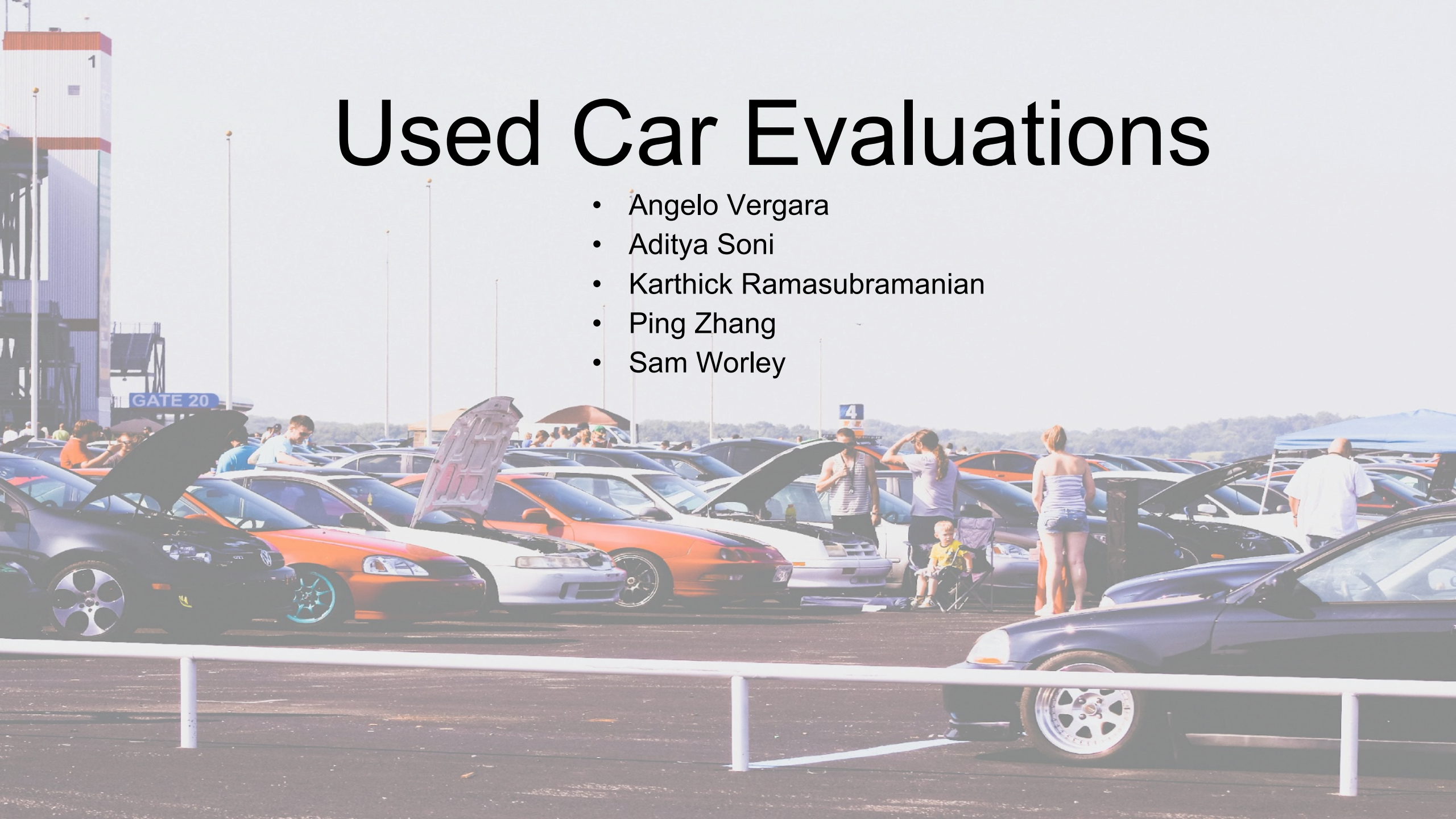


Used Car Evaluations

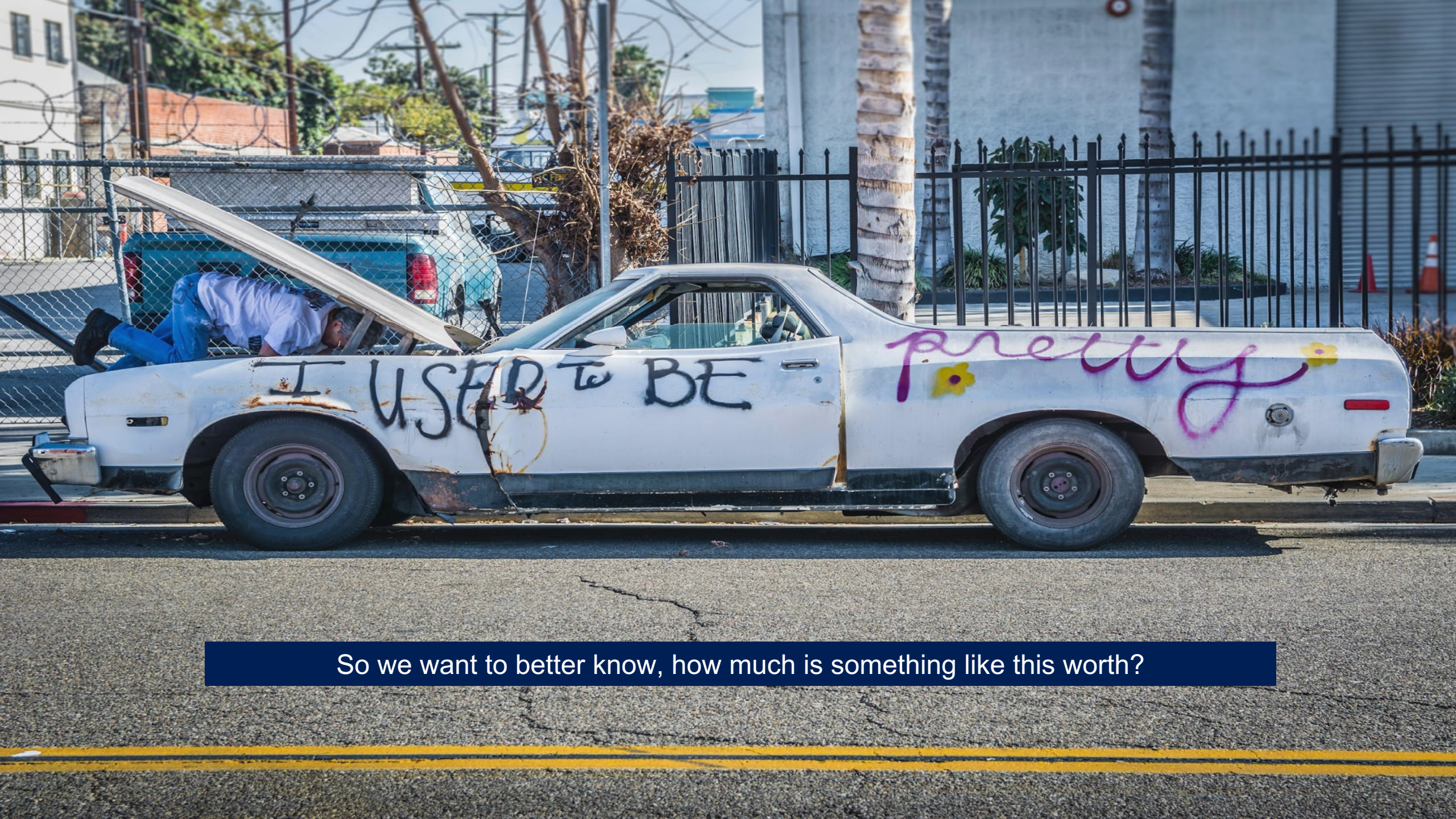
- Angelo Vergara
- Aditya Soni
- Karthick Ramasubramanian
- Ping Zhang
- Sam Worley

Opportunity

Evaluate car's worth

Make better offer

Improve Profits



So we want to better know, how much is something like this worth?

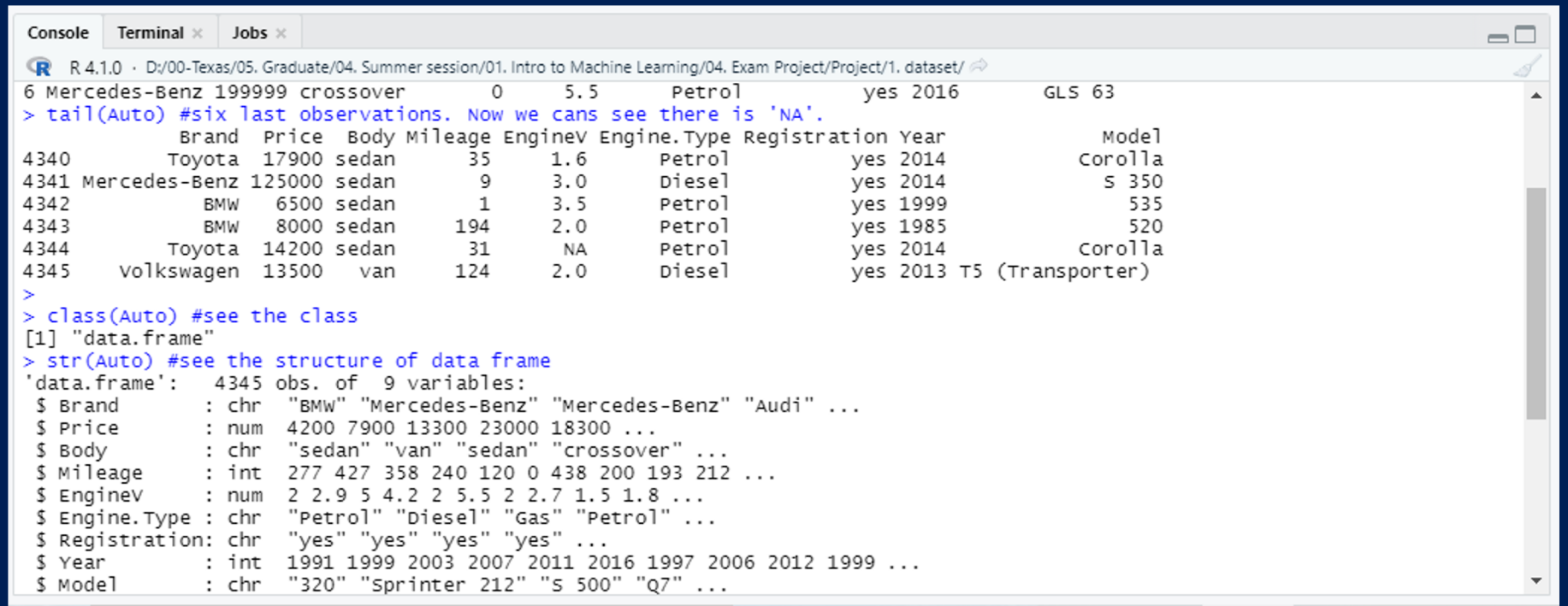
Exploratory Analysis

Exploratory Analysis

- Exploring the descriptive statistics of variables
  - Brand: Contains the various brands for Cars
  - Price: Contains the price of the Car
  - Body: Contains the name of main supporting of motor vehicle
  - Mileage: Contains the number of mileage of car
  - EngineV: Contains the engine version of Car
  - Engine Type: Contains the type of Engine
  - Registration: Show whether the car is registered or not
  - Year: Contains the year of manufacturing of Car
  - Model: Contains the model of Car

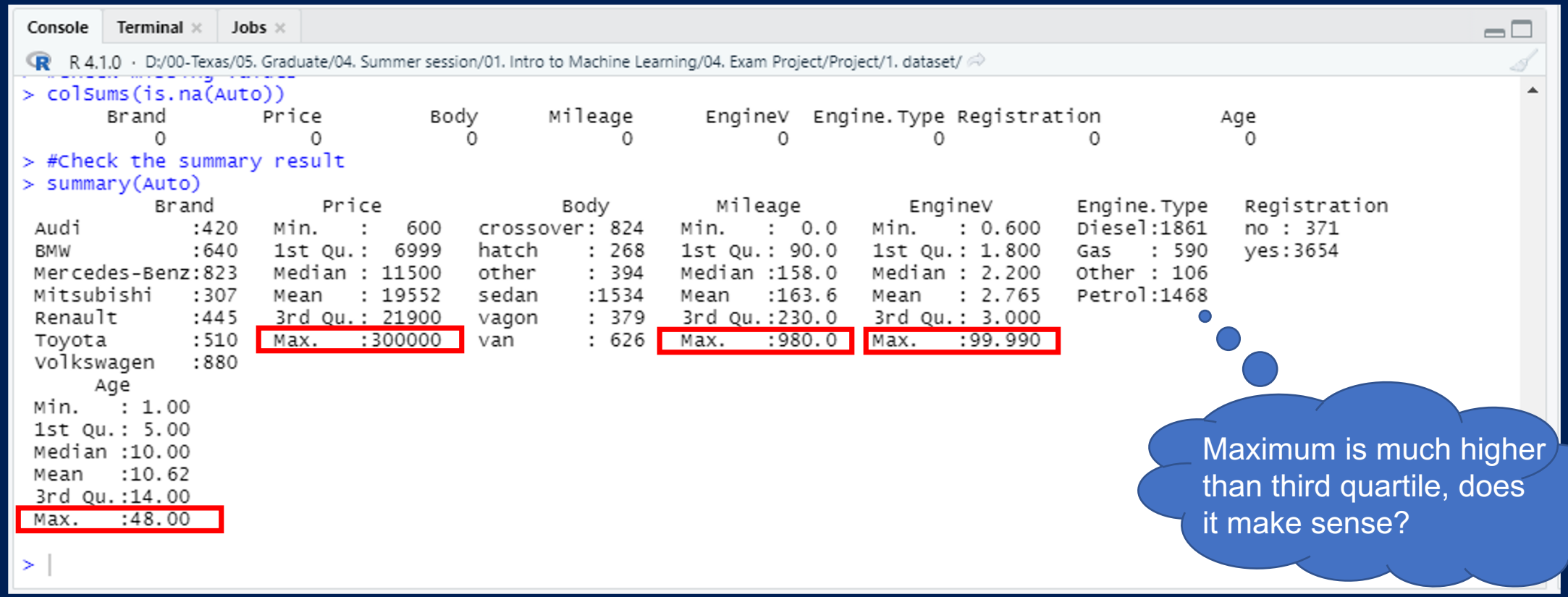
Think about:

- Model: 320 unique values, what will happen if we convert it into dummy variable?
- Year: Can we make it more sensible and readable?
- It shows that we have some 'NA' in the dataset, how shall we deal with them?



Exploratory Analysis

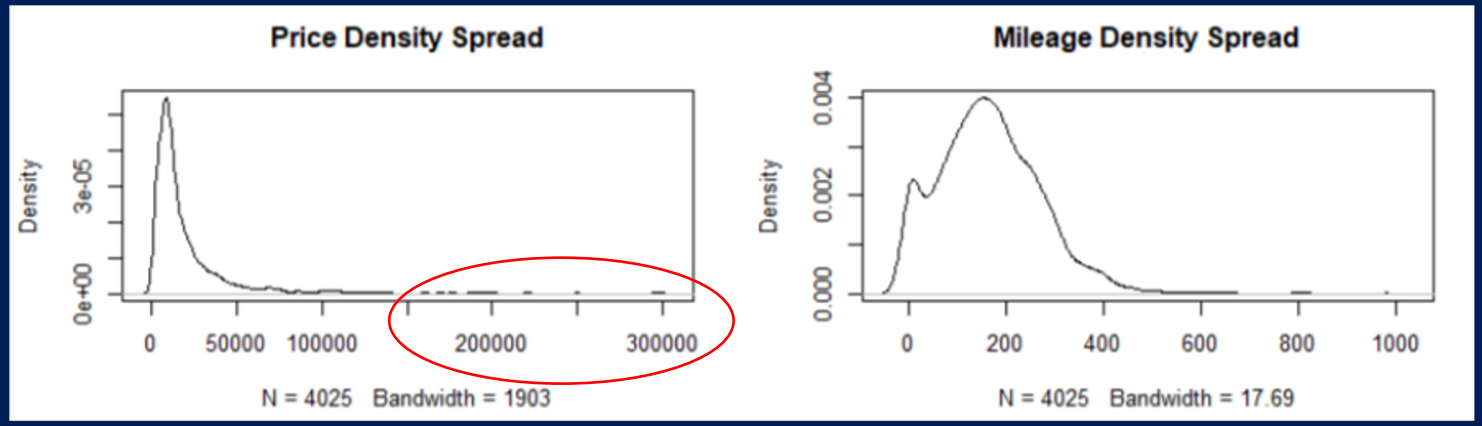
- Determining the variables of interest
  - Price ~ Brand + Body + Mileage + EngineV + Engine Type + Registration + Age
- Dealing with missing values
  - Exclude the observations with missing value

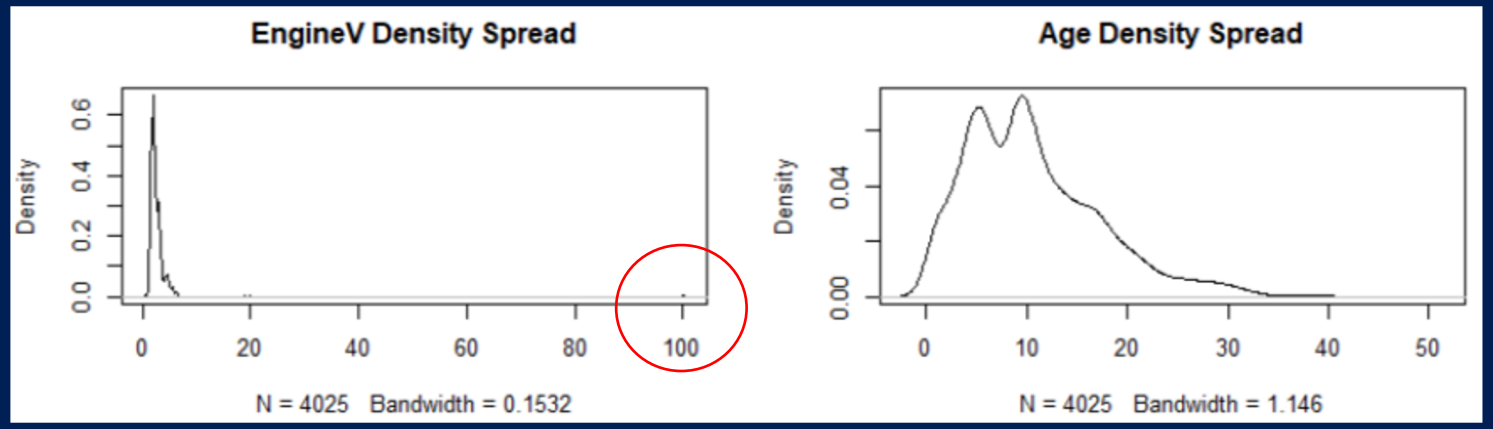


Exploratory Analysis

- Dealing with outliers
- Viewing trends in ‘Price’, ‘Mileage’, ‘EngineV’, ‘Age’, we can say that Price and EngineV column is not normally distributed, so we need to remove some outliers from data.

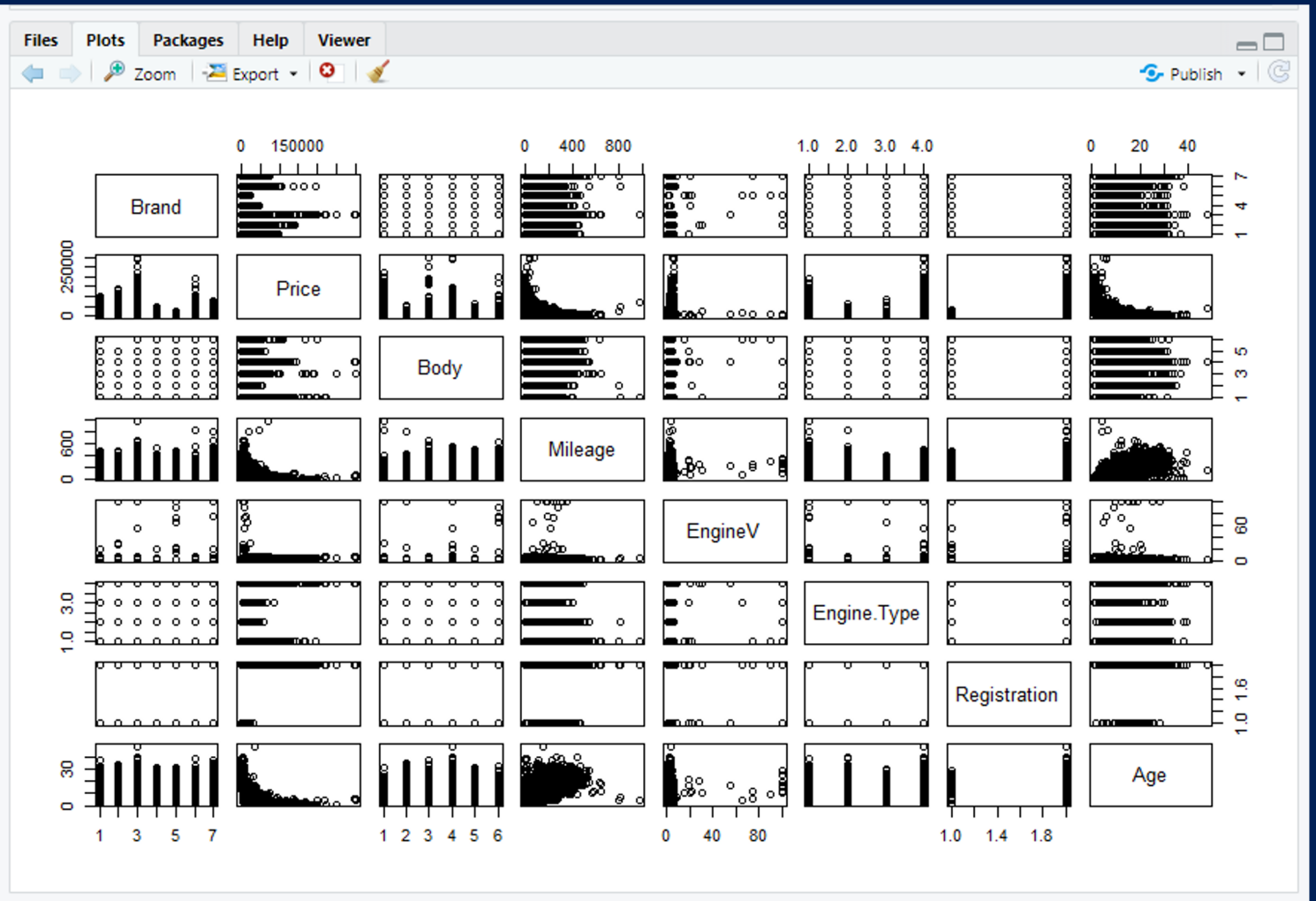
Think about:
- How to define outliers?
- How to handle with them?

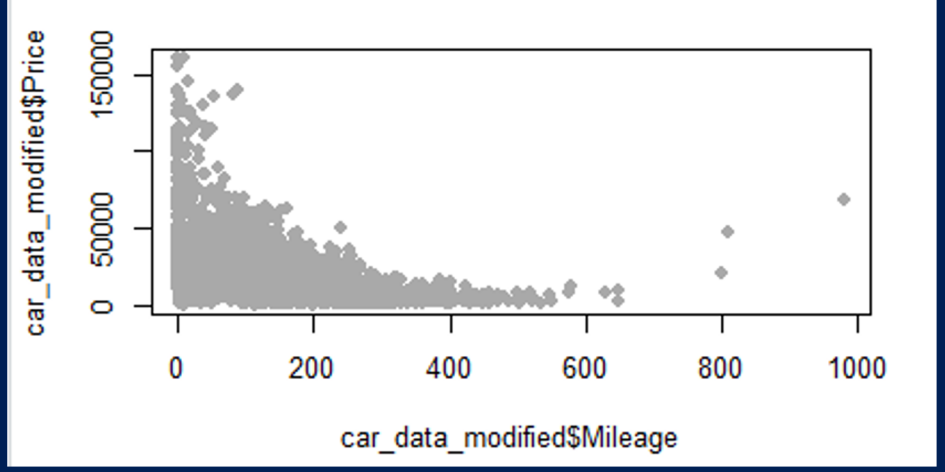


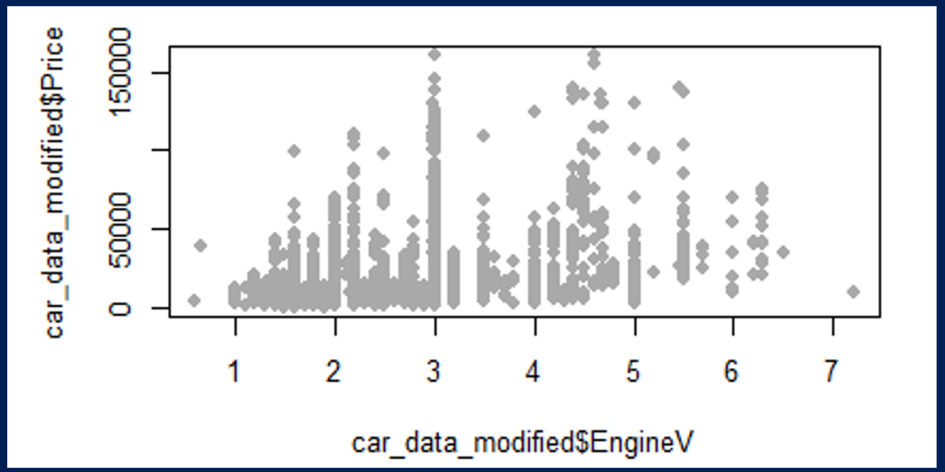


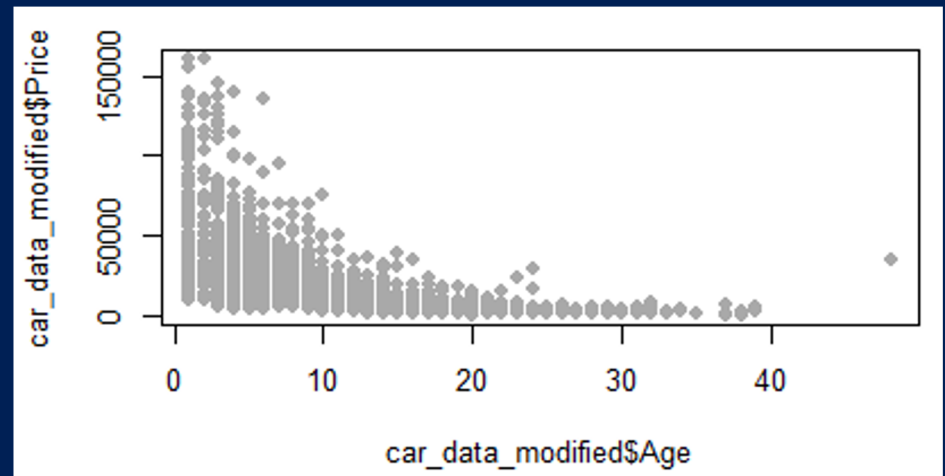
Exploratory Analysis

Getting a high understanding of potential relationship









Modeling

OLS

- Estimates make sense
  - BMW increases value
  - Mileage decreases value
- All predictors are significant according to their p-values

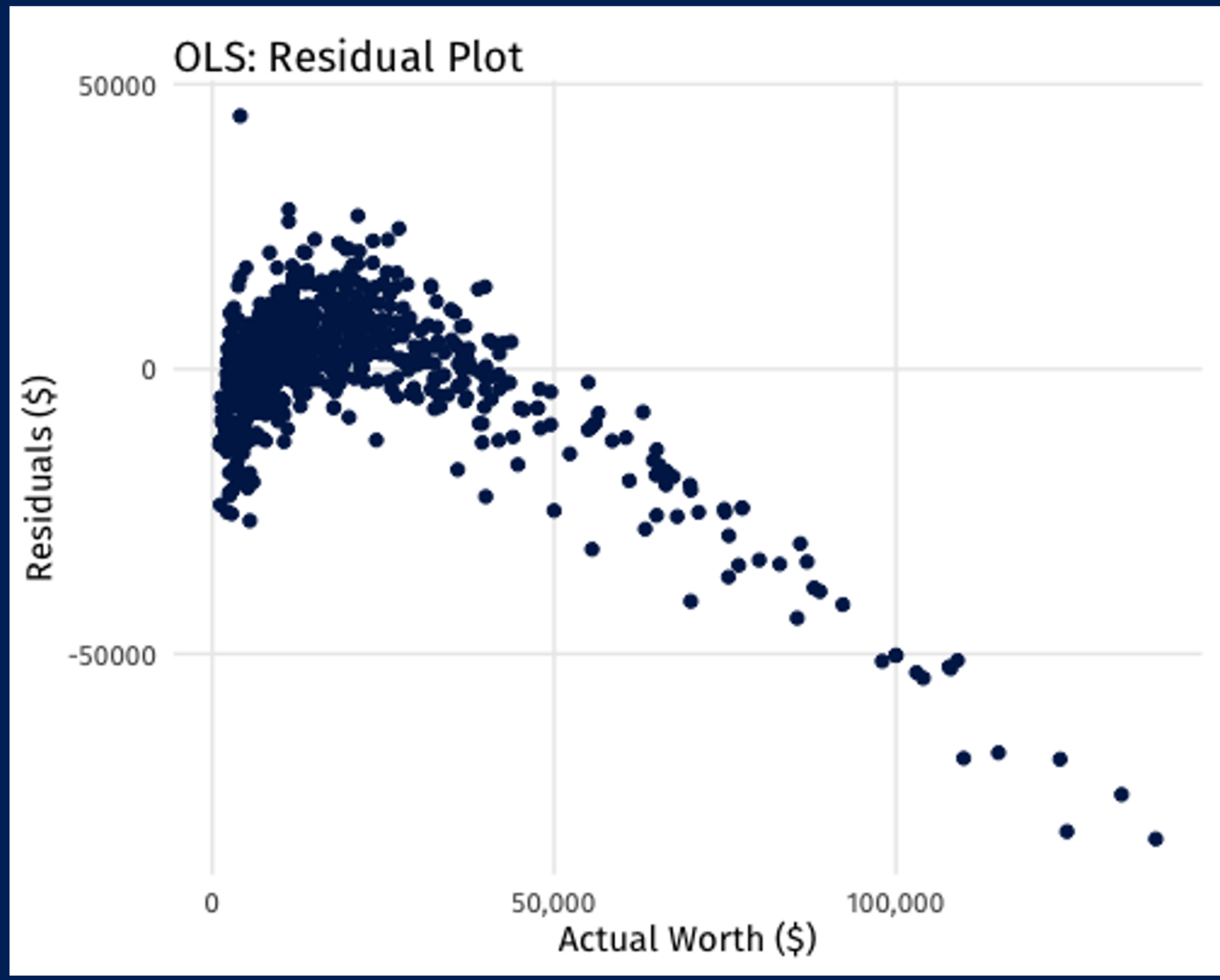
RMSE: 10147.52 ($)

term	estimate	p.value
(Intercept)	31947.6032	0
factor(Brand)BMW	2188.3049	0.0195
factor(Brand)Mercedes-Benz	4556.3168	0
factor(Brand)Mitsubishi	-12886.2956	0
factor(Brand)Renault	-10180.4874	0
factor(Brand)Toyota	-8382.303	0
factor(Brand)Volkswagen	-5040.7314	0
factor(Body)hatch	-10462.2235	0
factor(Body)other	-9968.0023	0
factor(Body)sedan	-10045.2522	0
factor(Body)vagon	-9920.78	0
factor(Body)van	-11320.7054	0
Mileage	-52.4754	0
EngineV	4618.1546	0
factor(Registration)yes	6463.9055	0
Age	-1028.5942	0

OLS

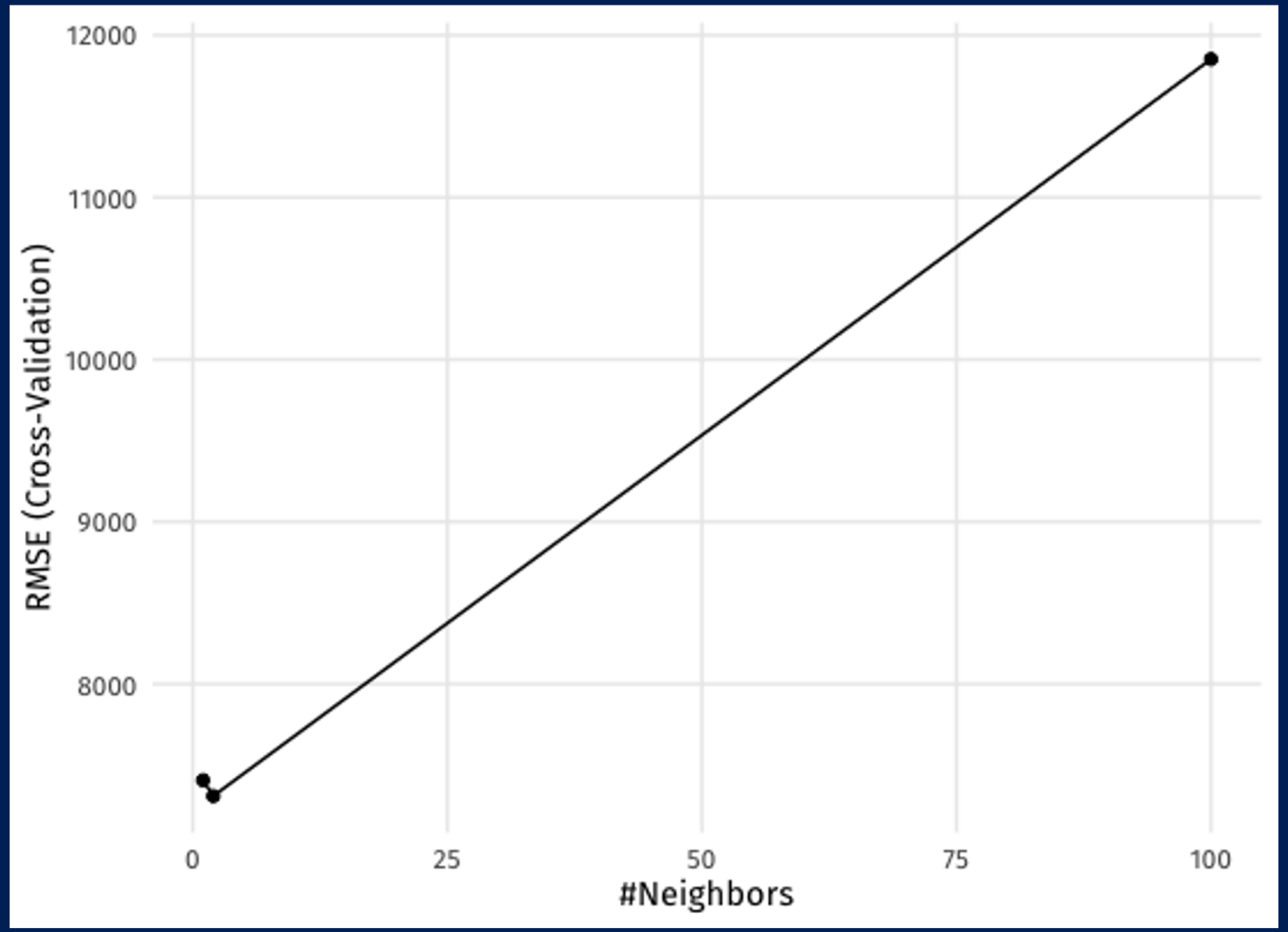
- Residuals – don’t look linear!
- Model underpredicts the more price increases
- Still a lot of variance even when price is lower

RMSE: 10147.52 ($)



KNN

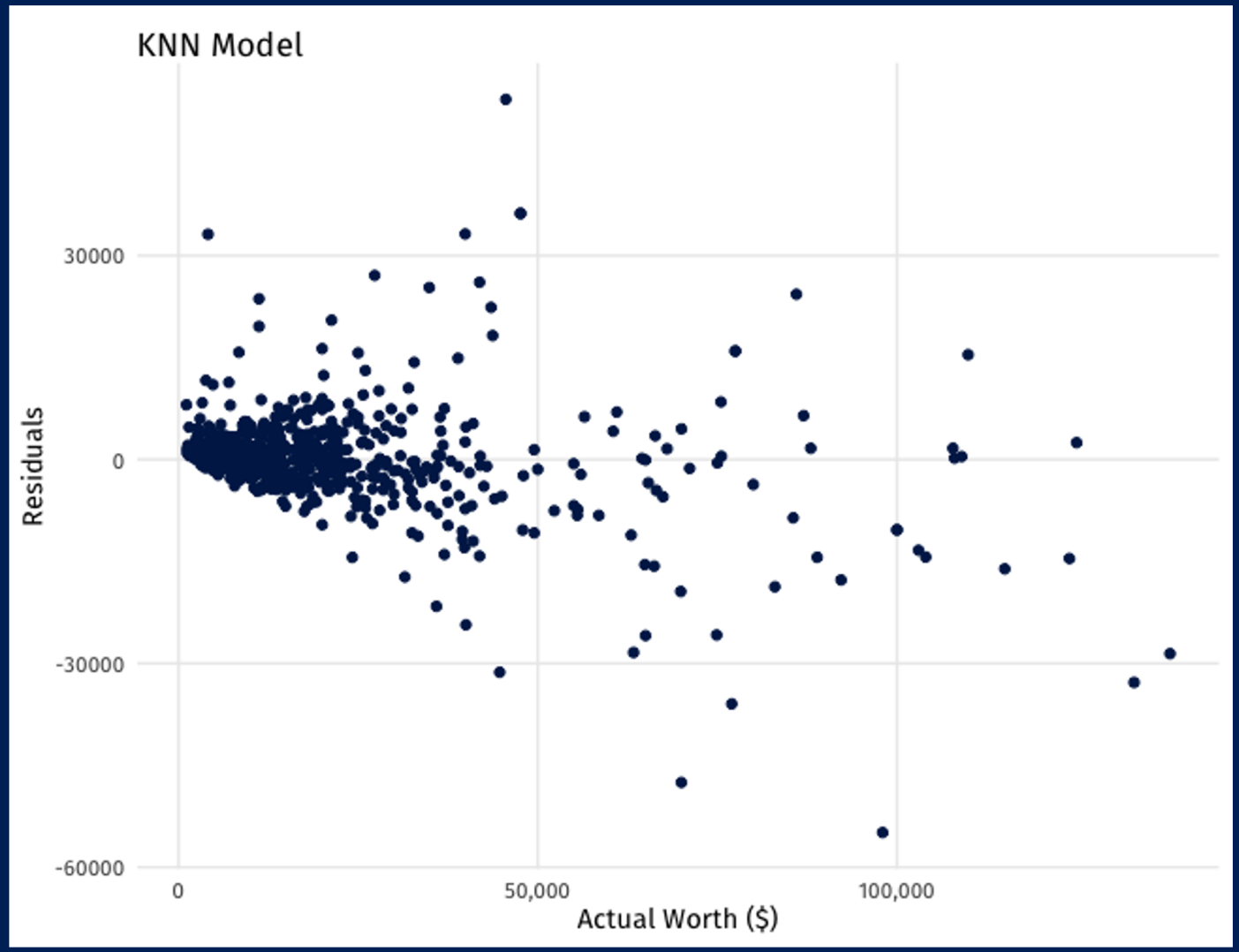
- Few # neighbors – high variance
- Still a lot of variance even when price is lower



RMSE: 7067 ($)

KNN

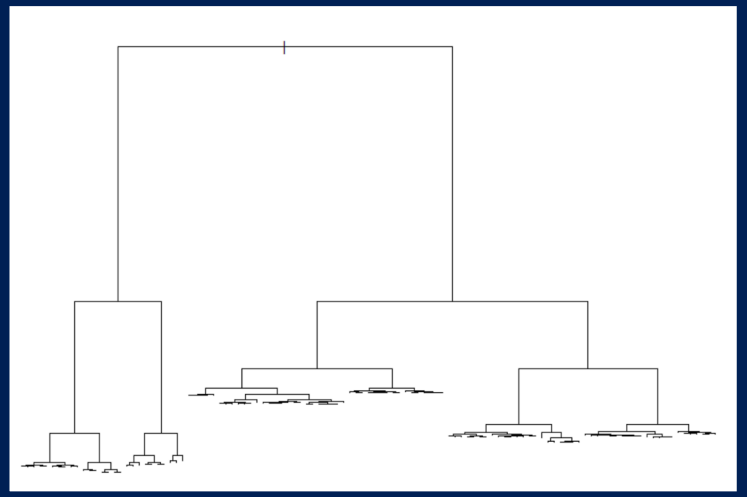
- Few # neighbors – high variance
- Still a lot of variance even when price is lower
- Tighter residuals

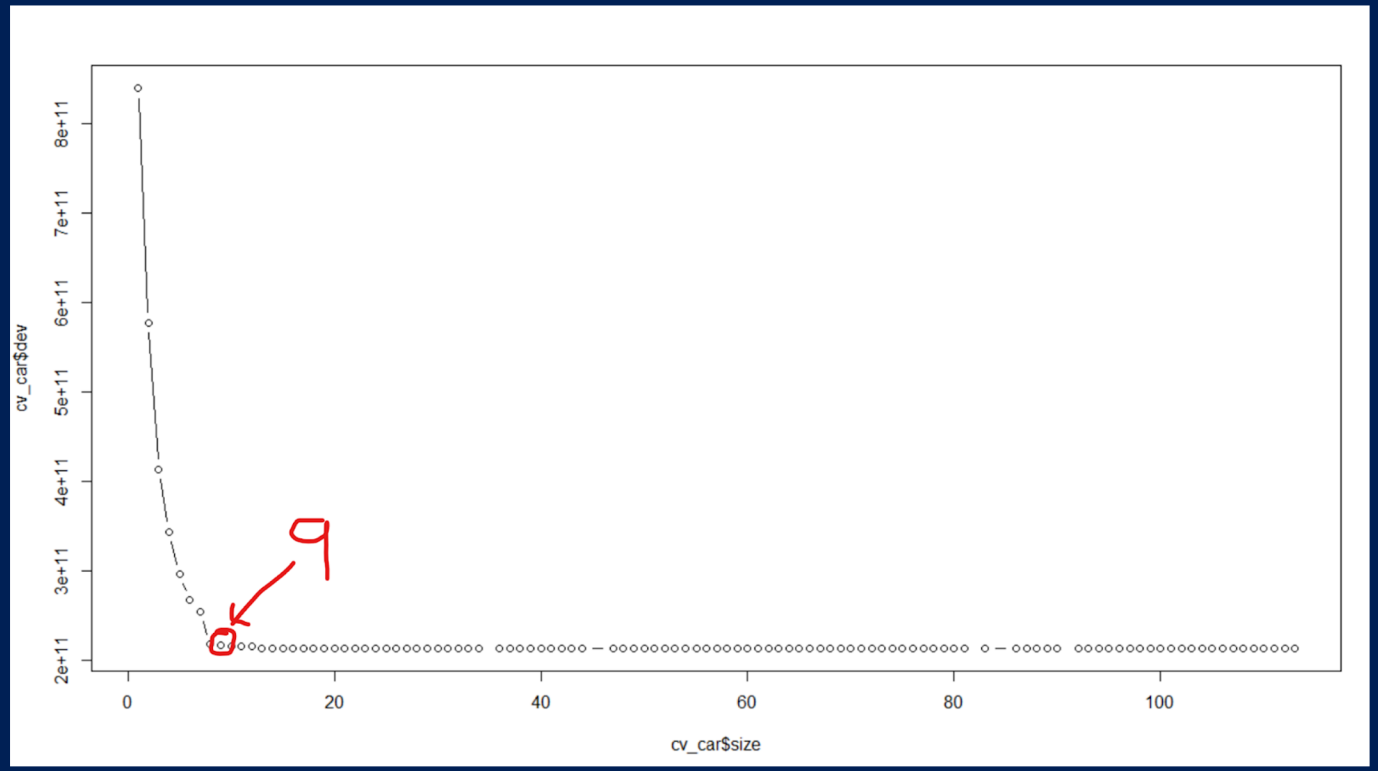


RMSE: 7067 ($)

Regression Tree

-Make a big tree
-Cross-validation





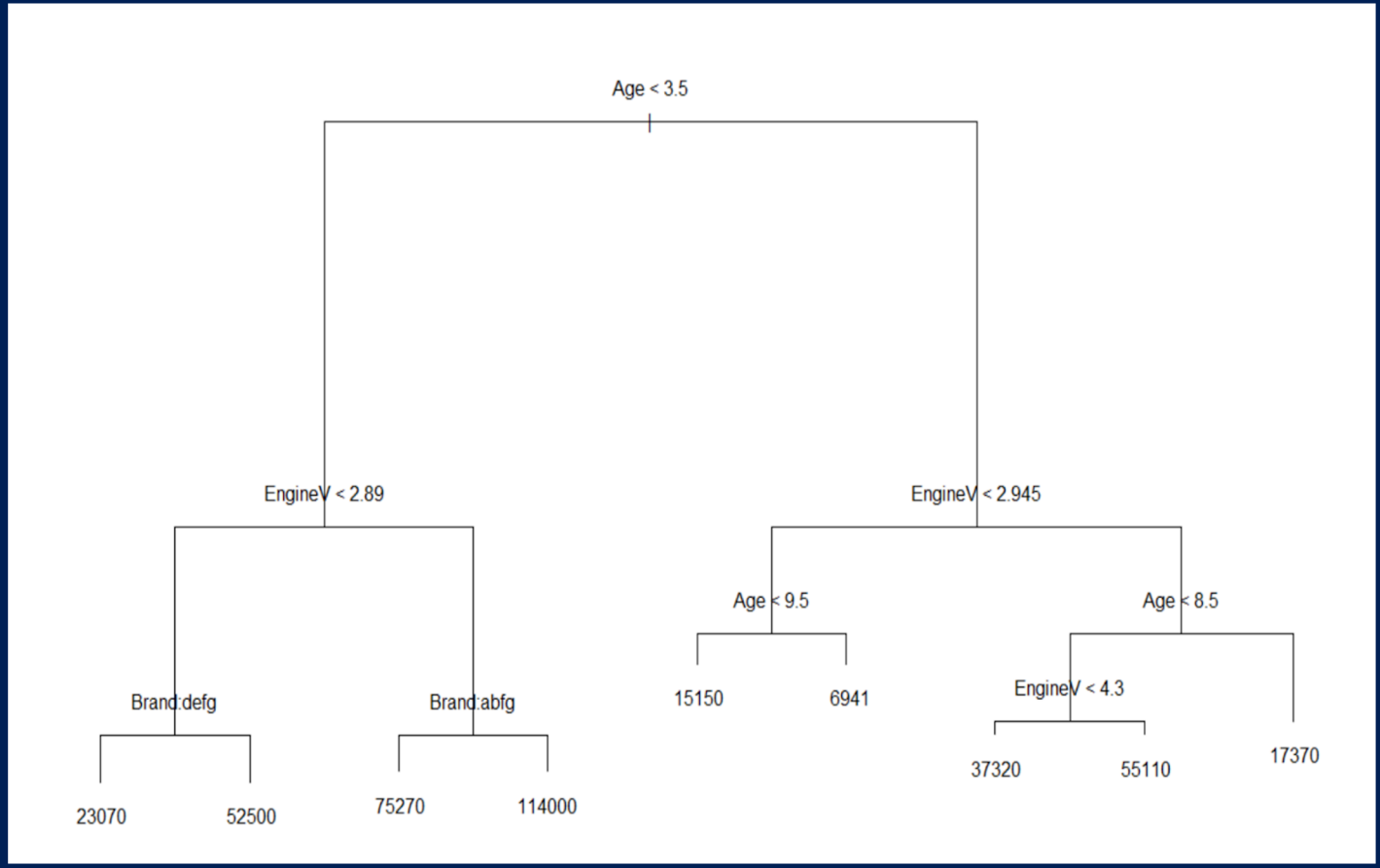
No. of nodes in the big tree = 113

After cross-validation, best model = 9 modes.

Regression Tree

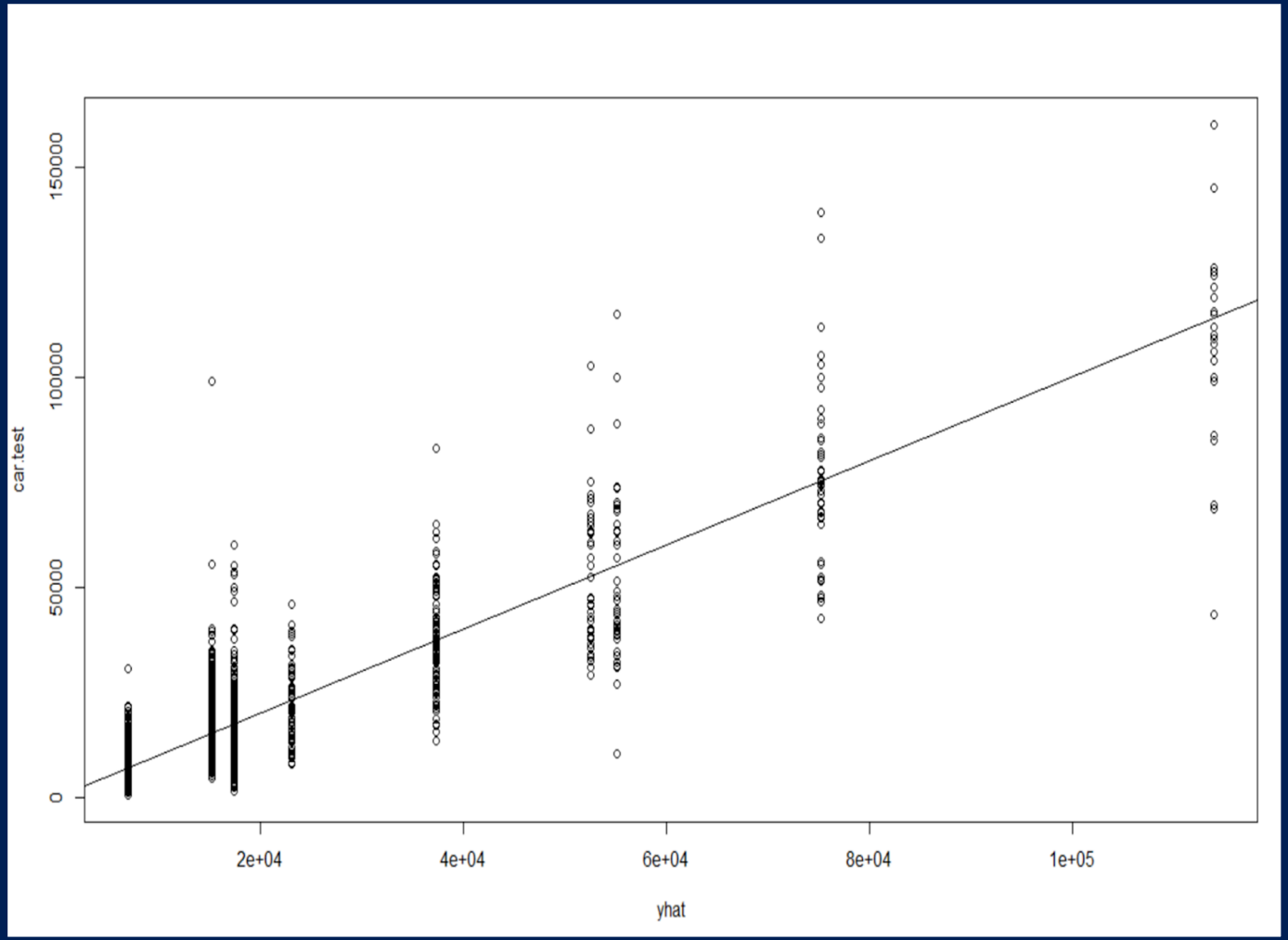
-Pruned to 9 terminal nodes

-Most important variables:
Age, Engine Volume, Brand



Regression Tree

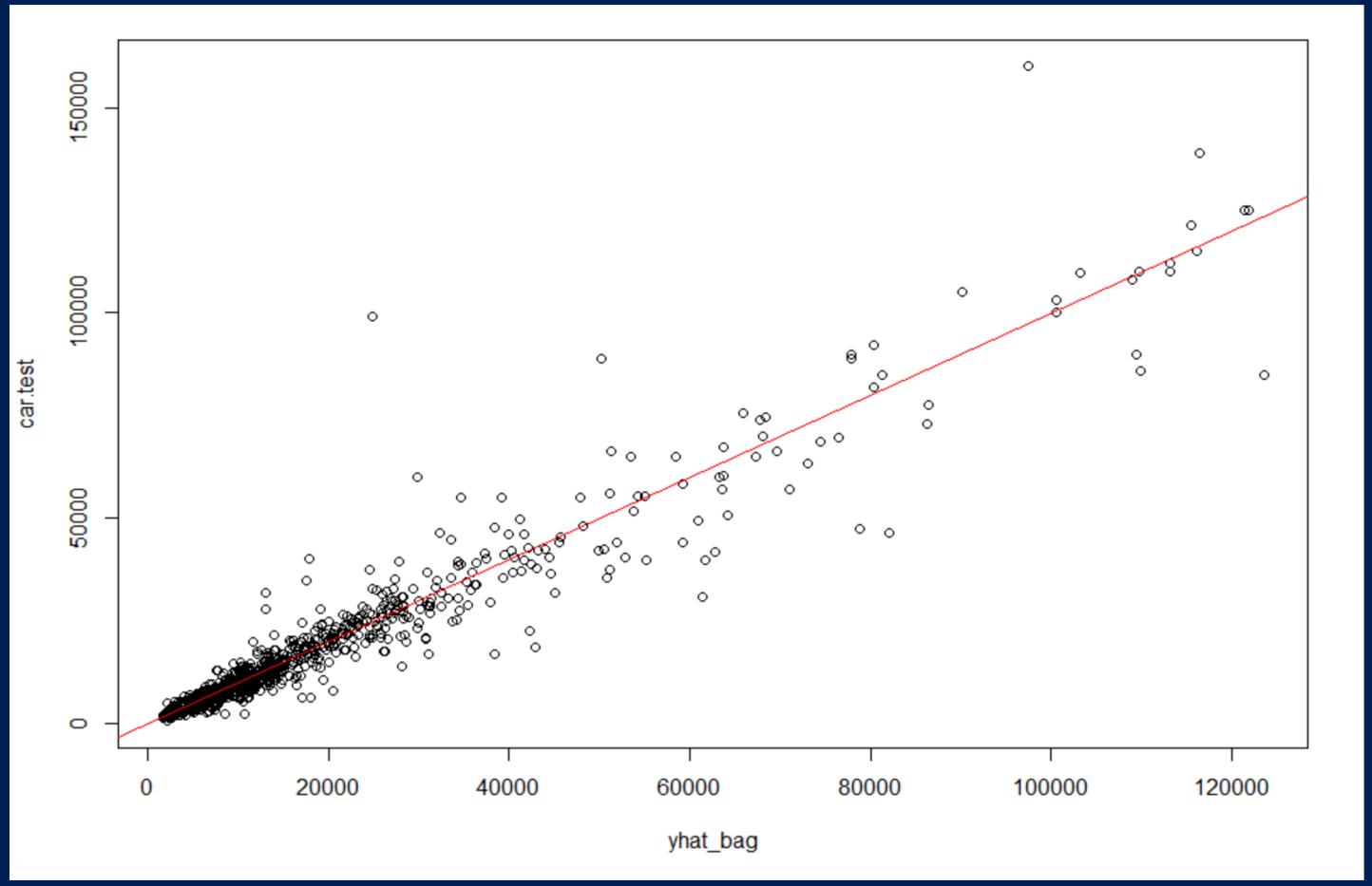
X-axis = Predicted prices
Y-axis = Test prices



RMSE: 9396 ($)

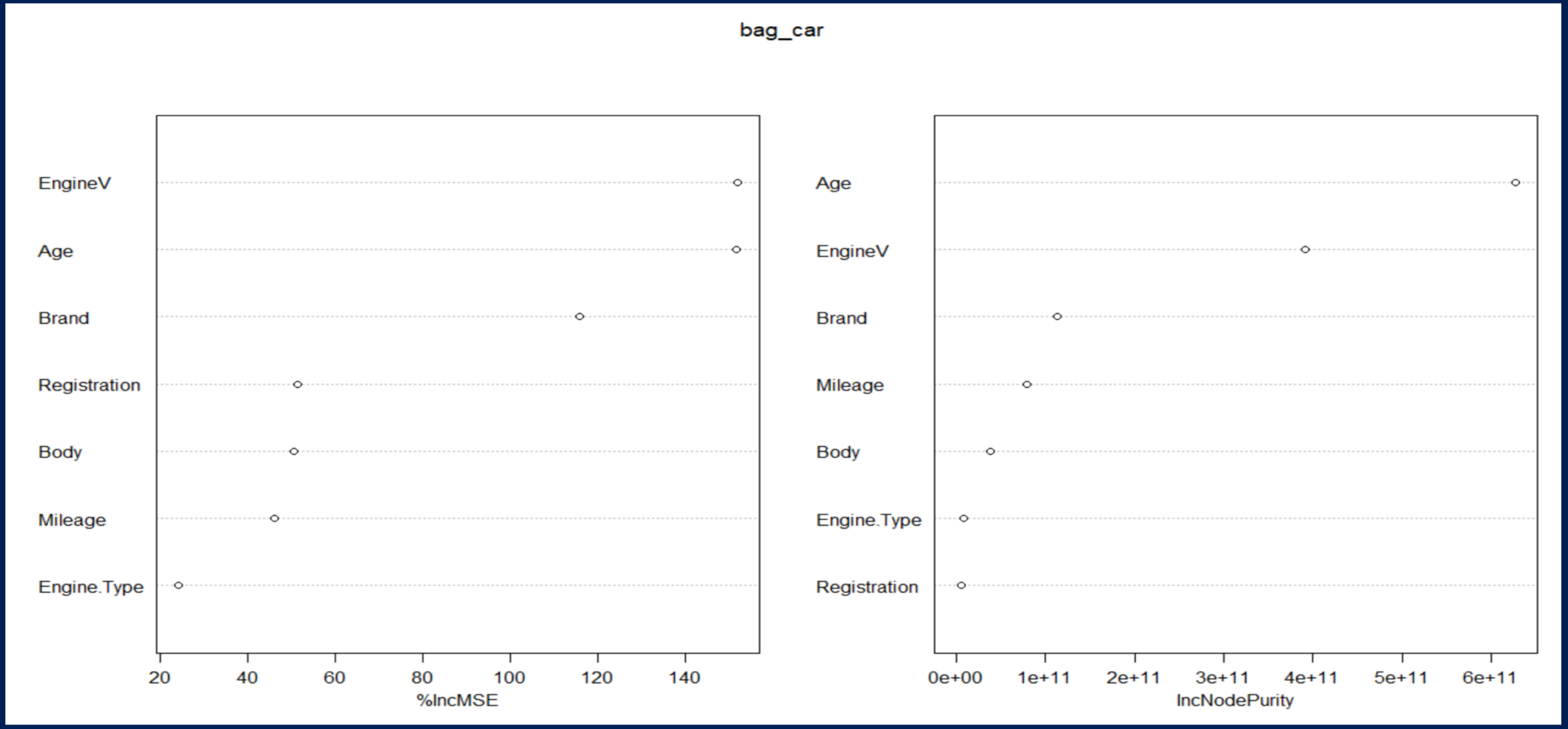
Bagging

X-axis = Predicted prices
Y-axis = Test prices



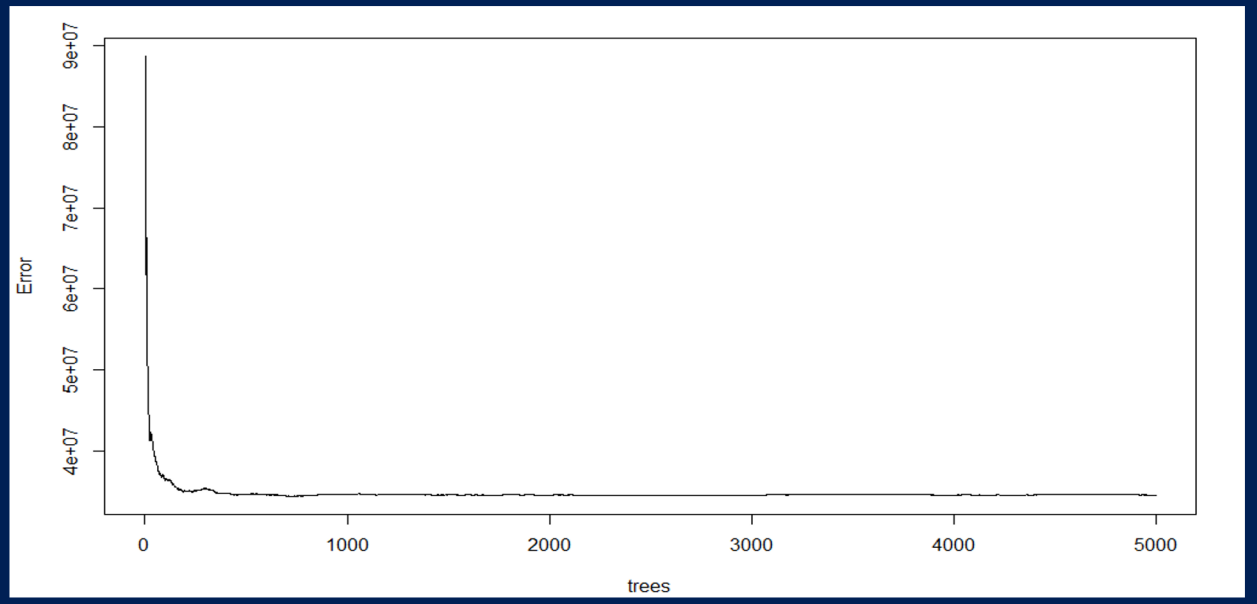
RMSE: 5985 ($)

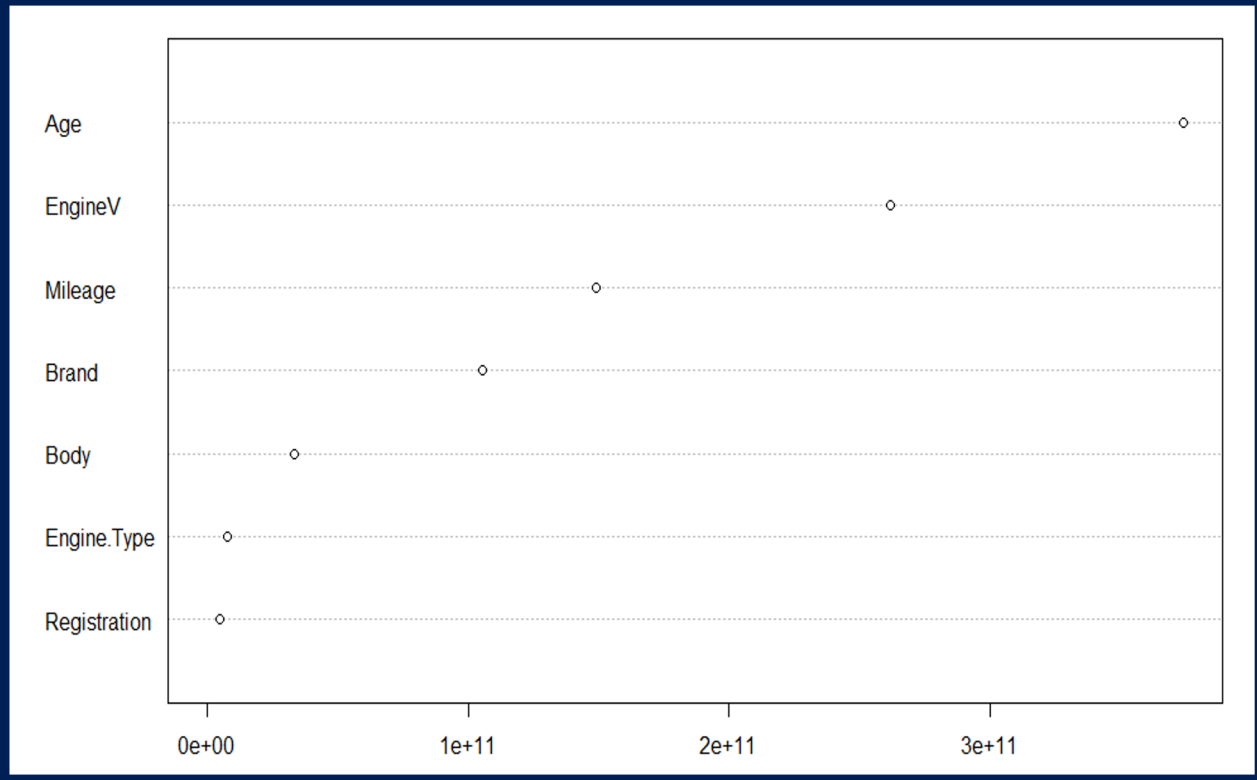
Bagging


bag_car
EngineV
Age
Brand
Registration
Body
Mileage
Engine.Type
20
40
60
80
100
120
140
%IncMSE
Age
EngineV
Brand
Mileage
Body
Engine.Type
Registration
0e+00
1e+11
2e+11
3e+11
4e+11
5e+11
6e+11
IncNodePurity

Random Forest

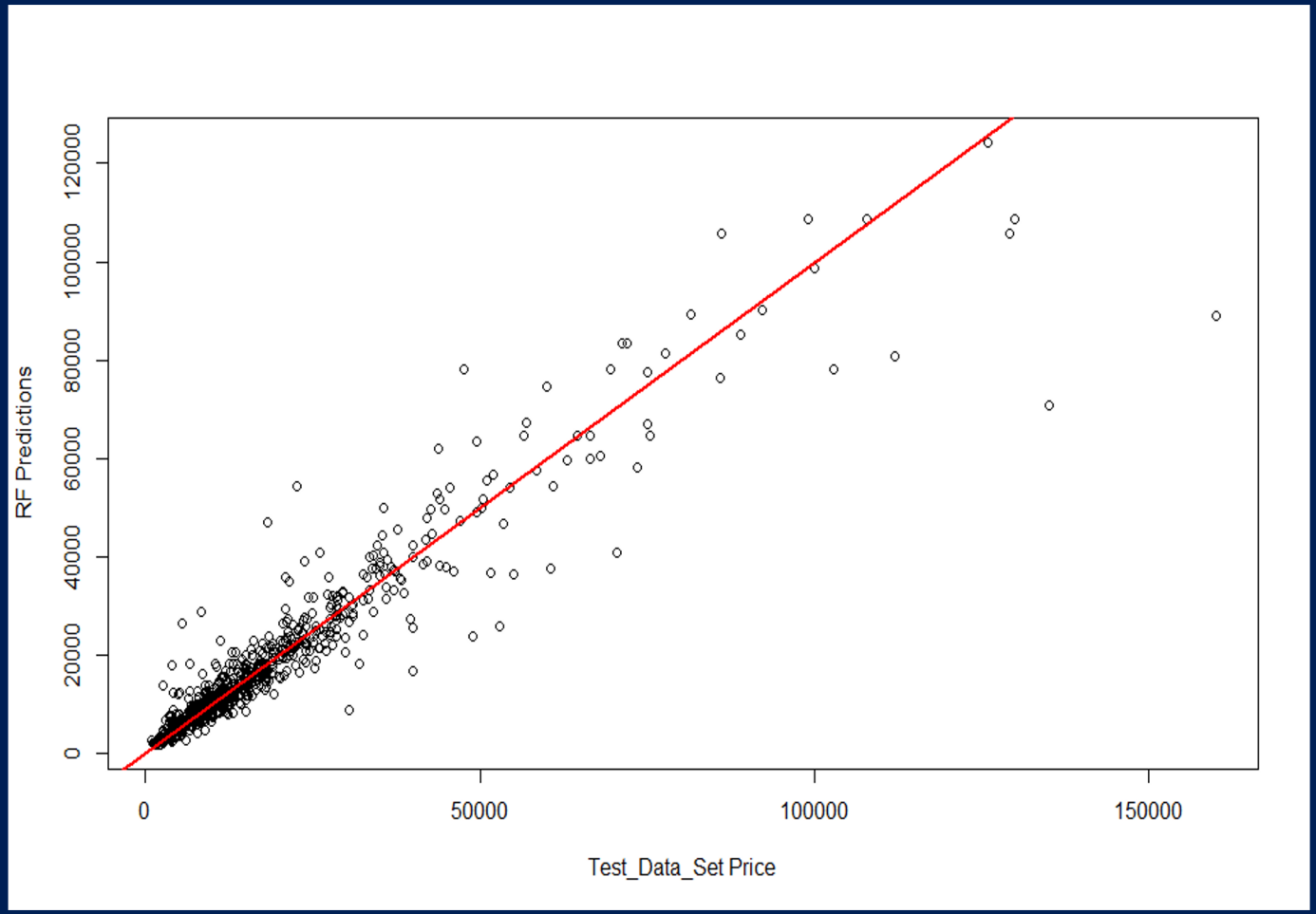
- Engine Type & Registration are significantly less important than other parameters.





Random Forest

- The best model is
  - All variables included
  - No transformations
- Overpredicting in lower values of price and underpredicting in higher price of Cars.



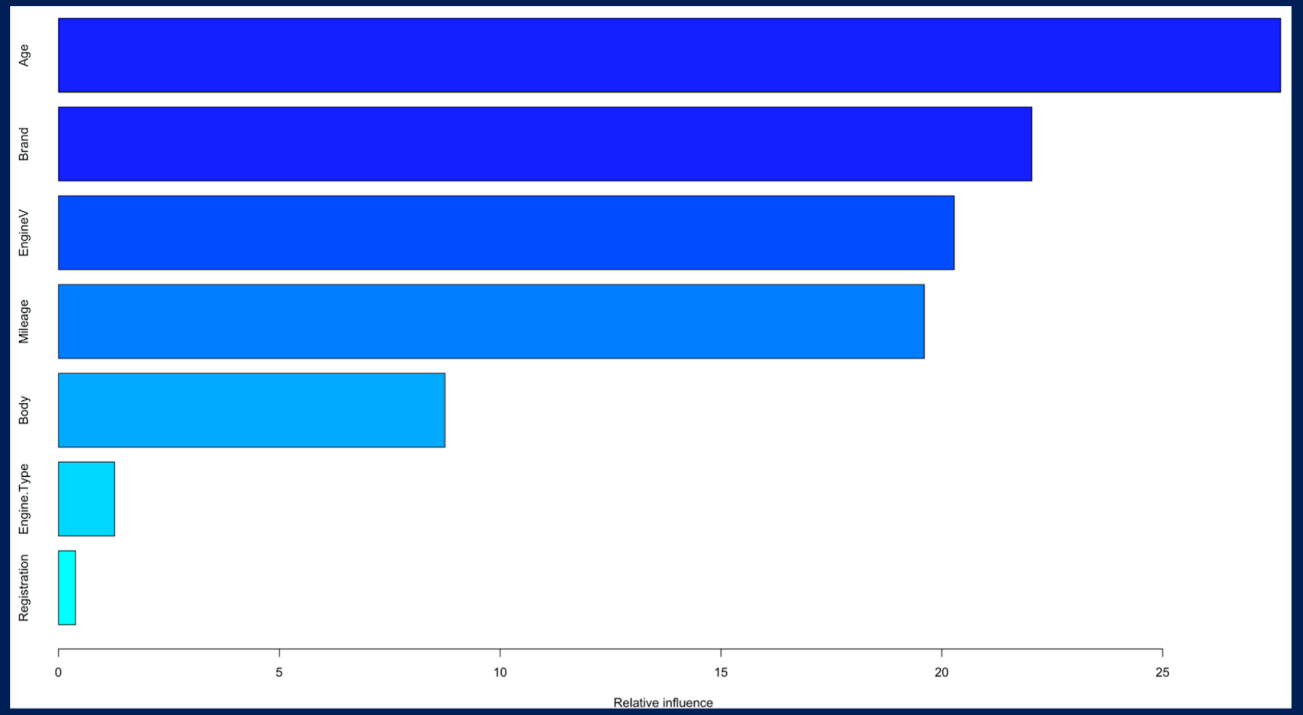
Model RMSE = $6178

Tree Size = 2000 and m = 5

Boosting

	tdepth	ntree	lam	olb	ilb
1	2	50	0.001	20383.859	19049.610
2	4	50	0.001	20229.369	18903.331
3	10	50	0.001	20103.918	18774.246
4	2	500	0.001	16978.746	15666.454
5	4	500	0.001	15831.509	14545.560
6	10	500	0.001	14835.252	13534.328
7	2	5000	0.001	7759.097	7111.937
8	4	5000	0.001	6814.893	6035.316
9	10	5000	0.001	6282.552	5115.845
10	2	50	0.200	7069.244	6536.577
11	4	50	0.200	6705.333	5669.159
12	10	50	0.200	6184.931	4660.043
13	2	500	0.200	6829.123	5318.588
14	4	500	0.200	6798.711	3895.426
15	10	500	0.200	6734.031	2670.951
16	2	5000	0.200	7623.399	3742.290
17	4	5000	0.200	7118.949	2392.951
18	10	5000	0.200	7340.658	1652.366

```
> which.min(olb)
[1] 12
```

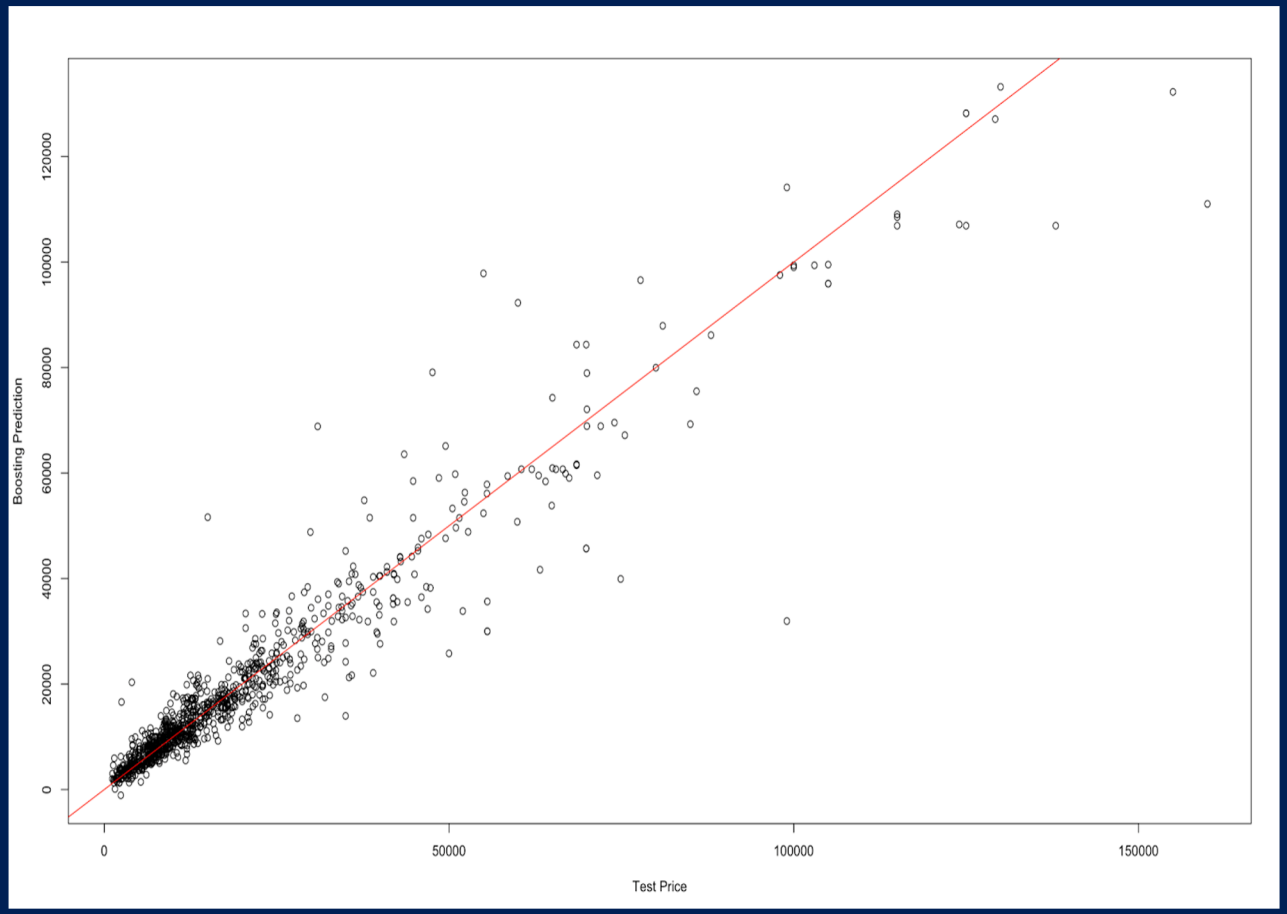


50 trees, each with 10 nodes,
with shrinkage .2

RMSE = 6184.9 ($)

Boosting

- Underpredicting higher valued cars
- Computed using optimal values with all predictors
- RMSE is good but not the best



RMSE = 6032 ($)

Conclusion

Model Accuracies

Models	RMSE
Bagging	$5,985
OLS	$10,148
KNN	$7,067
Regression Tree	$9,386
Boosting	$6,032
Random Forest	$6,178