PREDICT USED CAR PRICE

Capstone project

Problem Statement

- Due to the reduction of COE quota, we had seen an increase in the COE premium. As a result, more buyers are turning to the used car market. But the prices for a particular make and model can have a lot of difference in the used car market due to the a few reasons like:

  - How old is the car
  - Mileage
  - Number of owners etc

- For this project, we would like to train a model to predict the price of the car so as to let buyers have a guide on how much is the car that they are looking at

Work Flow

Sourcing for dataset

Understand the dataset and perform data cleaning

Exploratory Data Analysis

Data pre - processing and Feature Engineering

Model selection and training

Tuning of hyperparameters

Dataset

- Using the data that scrape from STCars website

Price	$115,800
Registration Date	29-Jul-2016
COE Remaining	5yrs 1mth 25days
Manufactured	2015
Mileage	87,600 km
No. of Owners	2
Transmission	Auto
Engine Capacity	1,984 cc
COE	$57,508
OMV	$40,292
Paper Value	$65,943
Depreciation	$17,770 / year
Type	Sports Car

Price	$69,800
Registration Date	15-Aug-2017
COE Remaining	6yrs 2mths 11days
Manufactured	2016
Mileage	47,000 km
No. of Owners	1
Transmission	Auto
Engine Capacity	1,498 cc
Fuel Type	Diesel
COE	$42,801
OMV	$25,671
Paper Value	$39,992
Depreciation	$9,810 / year
Type	SUV

- As can see that there is another row "Fuel Type" if the car is using diesel or petrol-electric. We have to take note of that when scraping the data.

Dataset

- A sample of our dataset

	0	1	2	3	4
make_model	MERCEDES-BENZ C-CLASS C200K (COE TILL 10/2028)	NISSAN X-TRAIL 2.0A PREMIUM 7-SEATER SUNROOF	NISSAN ELGRAND 2.5A HIGHWAY STAR	MERCEDES-BENZ CLS-CLASS CLS450 MILD HYBRID AMG...	TOYOTA HARRIER 2.4A G (COE TILL 09/2029)
Price	$63,800	$81,800	$88,000	$367,988	$58,500
Registration Date	19-Dec-2008	21-Mar-2017	03-Aug-2016	30-Oct-2020	23-Sep-2009
COE Remaining	7yrs 5mths 13days	5yrs 10mths 2days	5yrs 2mths 15days	9yrs 5mths 11days	8yrs 4mths 4days
Manufactured	2008	2016	2016	2020	2008
Mileage	165,000 km	74,000 km	82,225 km	652 km	174,000 km
No. of Owners	4	1	1	1	5
Transmission	Auto	Auto	Auto	Auto	Auto
Engine Capacity	1,796 cc	1,997 cc	2,488 cc	2,999 cc	2,362 cc
fuel_type	NaN	NaN	NaN	Petrol-Electric	NaN
COE	$32,279	$53,300	$57,501	$32,801	$37,941
OMV	$43,106	$23,955	$34,974	$83,552	$27,778
Paper Value	$24,081	$50,286	$60,687	$122,808	$31,694
Depreciation	$8,550 / year	$11,810 / year	$12,950 / year	$32,440 / year	$7,000 / year
Type	Luxury Sedan	SUV	MPV	Luxury Sedan	SUV

Dataset

- Consist of 7244 rows and 15 features:

  1. make_model – brand and model of the car
  2. Price – price of the car (our label)
  3. Registration Date – first registration date of the car
  4. COE Remaining – remaining COE of the car
  5. Manufactured – year the car is manufactured
  6. Mileage – mileage of the car in km
  7. No. of Owners – car is owned by how many owner before

Dataset

8. Transmission – auto or manual gear
9. Engine Capacity – engine capacity of the car in c.c
10. fuel_type – petrol, diesel, petrol – electric
11. COE – COE premium paid for the car when first registered
12. OMV – open market value of the car
13. Paper Value – the amount you get if the car is deregistered
14. Depreciation – how much the car depreciate per year
15. Type – type of the car (luxury sedan, MPV, SUV etc)

Dataset

- Features that we think could be important:

Registration Date

- It can tell us how old is the car

Mileage

- It can tell us how often the car is been driven

COE Remaining

- It can tell us how long more can the car be driven on road

Depreciation

- It can tell us how much we will lose per year

Questions to ask ourself

- Machine Learning (Supervised Regression Model) :

  I. Are we going to create or remove any features that may affect the model performance

  II. How are we going to deal with the outliers and missing values

  III. What to do if the feature distribution is skewed

  IV. Which encoding method to use for categorical features

  V. How many baseline models are we going to train and perform hyperparameter tuning

Data Cleaning

1. Any duplicate

```
Duplicated data: 2015
```

2. "NaN" values

```
make_model               0
Price                    0
Registration Date        0
COE Remaining            0
Manufactured             0
Mileage                  0
No. of Owners            0
Transmission             0
Engine Capacity          0
fuel_type             4700
COE                      0
OMV                      0
Paper Value              0
Depreciation             0
Type                     0
```

1. Drop all and keep first
2. Replace with "Petrol"

1. After dropping

```
df.shape
(5229, 15)
```

2. After replacing

```
make_model               0
Price                    0
Registration Date        0
COE Remaining            0
Manufactured             0
Mileage                  0
No. of Owners            0
Transmission             0
Engine Capacity          0
fuel_type                0
COE                      0
OMV                      0
Paper Value              0
Depreciation             0
Type                     0
```

3. Convert features dtype

```
make_model           object
Price                object
Registration Date    object
COE Remaining        object
Manufactured         int64
Mileage              object
No. of Owners        object
Transmission         object
Engine Capacity      object
fuel_type            object
COE                  object
OMV                  object
Paper Value          object
Depreciation         object
Type                 object
```

Convert these features to integer

```
make_model           object
Price                int64
Registration Date    object
COE Remaining        int64
Manufactured         int64
Mileage              int64
No. of Owners        int64
Transmission         object
Engine Capacity      int64
fuel_type            object
COE                  int64
OMV                  int64
Paper Value          int64
Depreciation         int64
Type                 object
electric             object
```

Data Cleaning

4. **Engine Capacity**

```
177       Electric
2149      Electric
829       Electric
2475      Electric
850       Electric
             ...
3365      1,193 cc
3599      1,193 cc
1175      1,086 cc
2373      1,086 cc
4960      1,086 cc
Name: Engine Capacity,
```

Create new feature “electric” to show if the car is electric driven

```
0      No
1      No
2      No
3      No
4      No
Name: electric
```

5. **make_model**

	make_model
0	MERCEDES-BENZ C-CLASS C200K (COE TILL 10/2028)
1	NISSAN X-TRAIL 2.0A PREMIUM 7-SEATER SUNROOF

Create another feature “make” to show the brand

	make	model
0	MERCEDES-BENZ	MERCEDES-BENZ C-CLASS C200K (COE TILL 10/2028)
1	NISSAN	NISSAN X-TRAIL 2.0A PREMIUM 7-SEATER SUNROOF

6. **Registration Date**

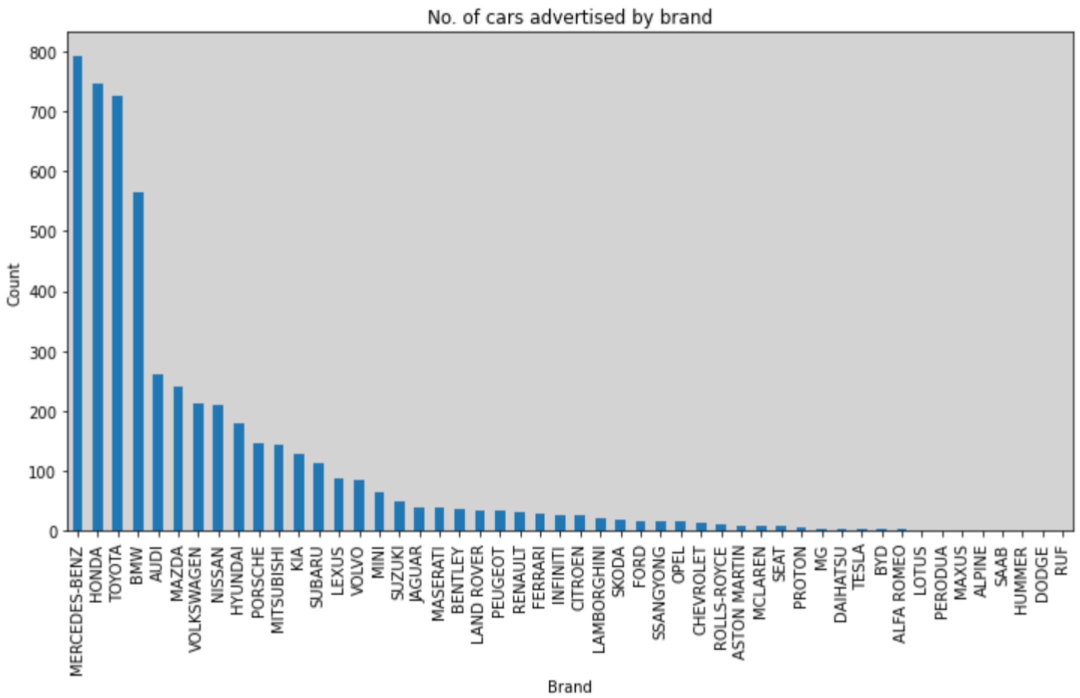
Registration Date
19-Dec-2008
21-Mar-2017
03-Aug-2016

Replace with new feature “car_age”

car_age
13
4
5

Exploratory Data Analysis

1. What car brand are advertised on the website



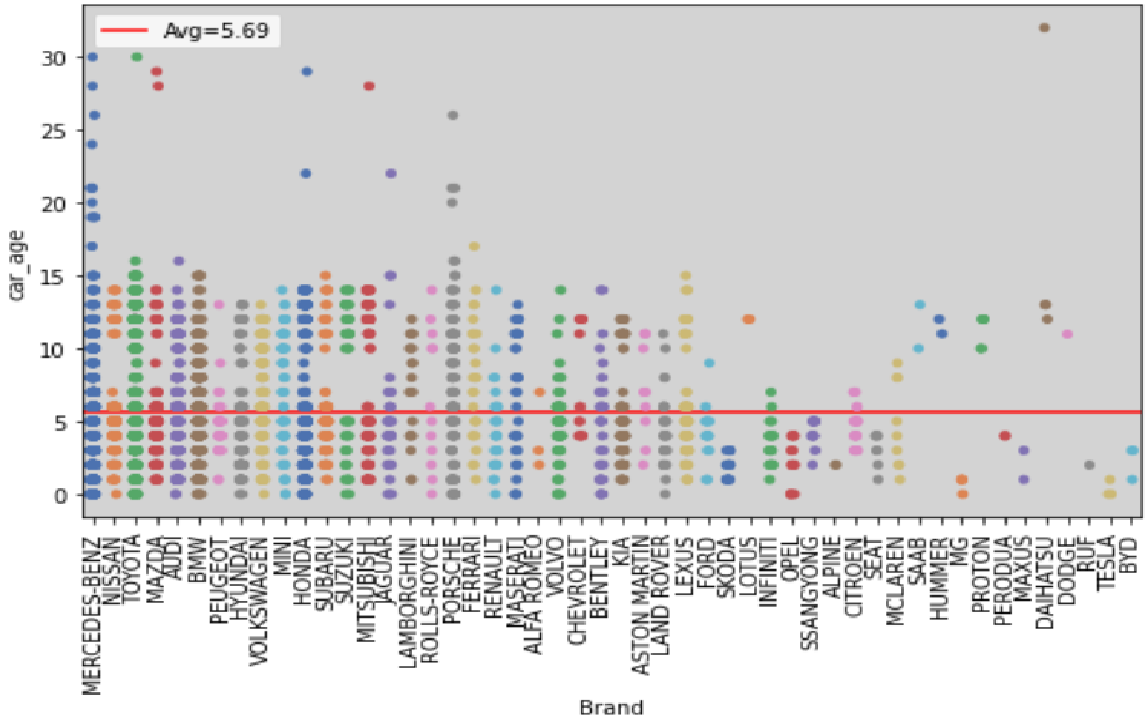
Left - Mercedes, Honda, Toyota and BMW combined to have more than half of the total cars advertised

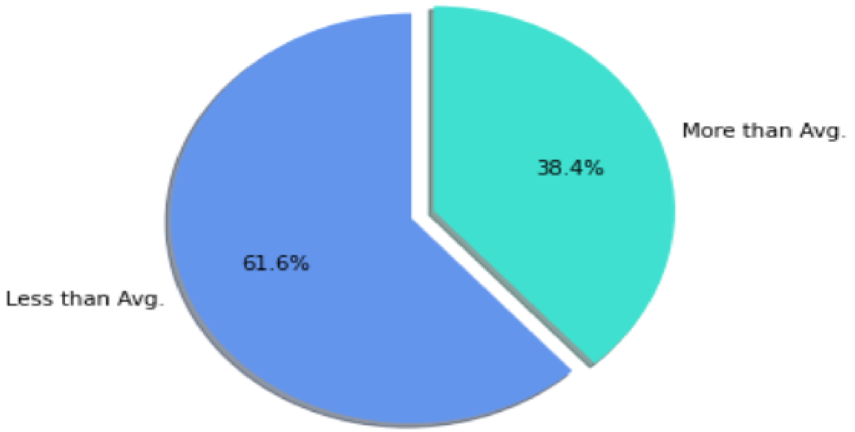
Right - These 4 brands already had a huge market shares since 2005
- Statistics from data.gov.sg: Annual car population by make

Exploratory Data Analysis

2. Car age distribution of each brand

Car Age





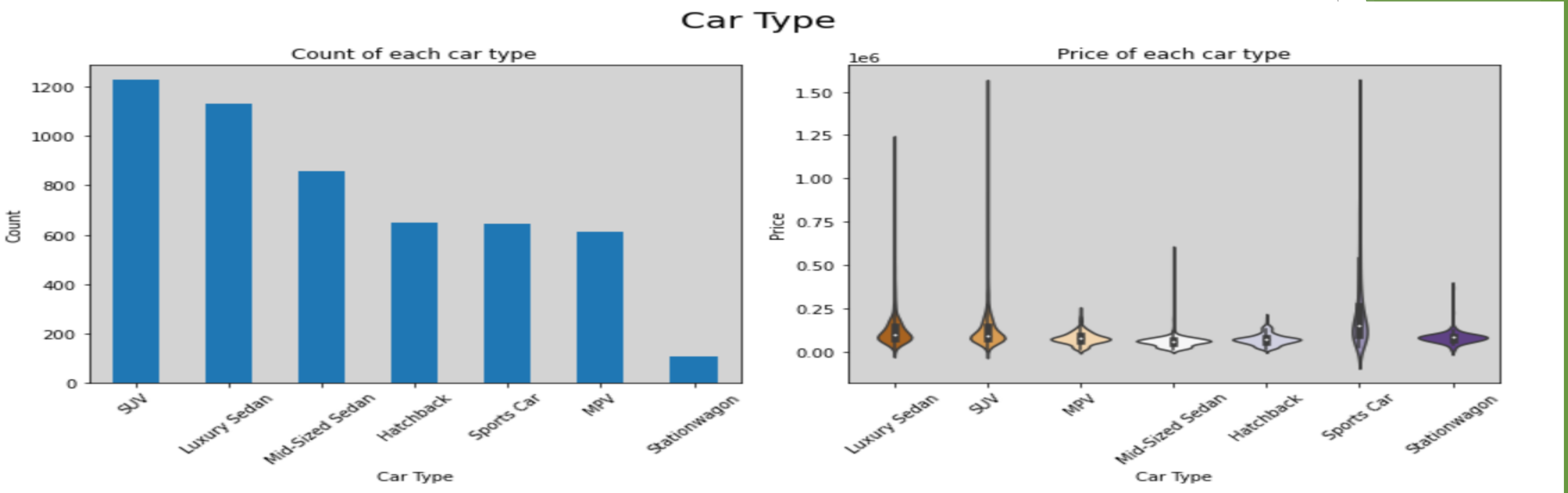
Left - The age distribution of most brand are very wide. There are even vintage cars (more than 20 years)

- The average age of the cars on the website are below 6 year

Right - Most of them are below the average.

Exploratory Data Analysis

3. Which car type are more popular

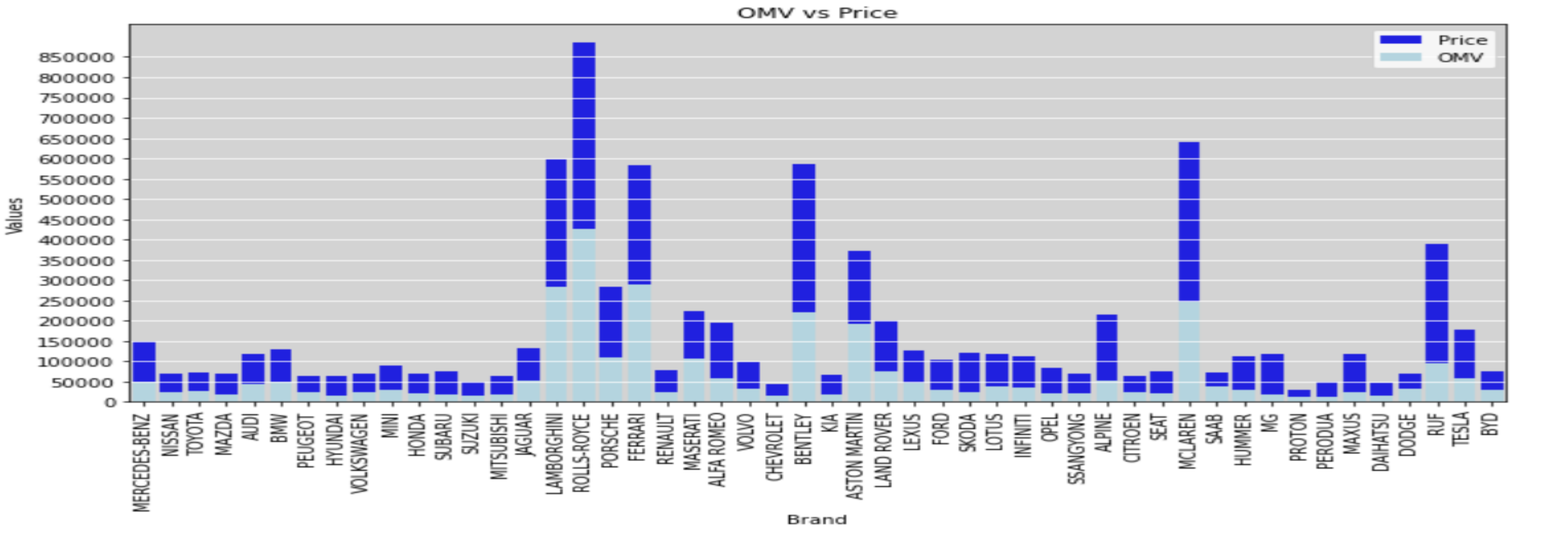


Left - SUV is the second most common type of car on the website (Luxury Sedan and Mid-Size Sedan are still sedan car. Difference is just the brand)
- Demand for SUV went up?

Right - Due to the brand (like Roll-Royces), price of some car types can go as high as over $1 millions (for a used car?!)
- The highest price for Mid-Sized Sedan is above $500k (Seems like some of the car type are mixed up)

Exploratory Data Analysis

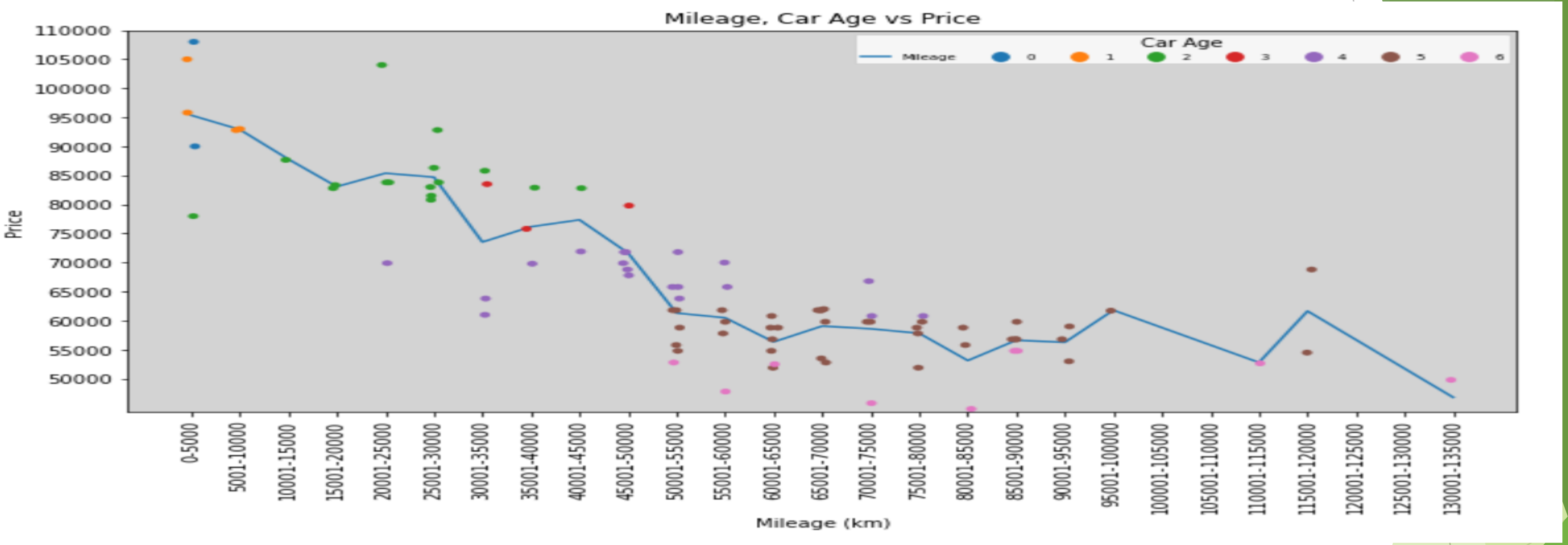
4. What can OMV tell us about the car



- We can see that those cars with **OMV** of $100k and above are those exotic brands like Rolls-Royce, Ferari, Lamborghini etc
- In some way **OMV** can roughly tell us if the car is from an exotic brand
- We can see that car price is about at least 1.5 times the OMV of the car.

Exploratory Data Analysis

5. Does mileage and age affect the car pricing (using 1 brand and model)



- We can see that most of the car price is highly affected by car age and mileage
- The price is lower if the car is older or with higher mileage.

Data Pre-processing

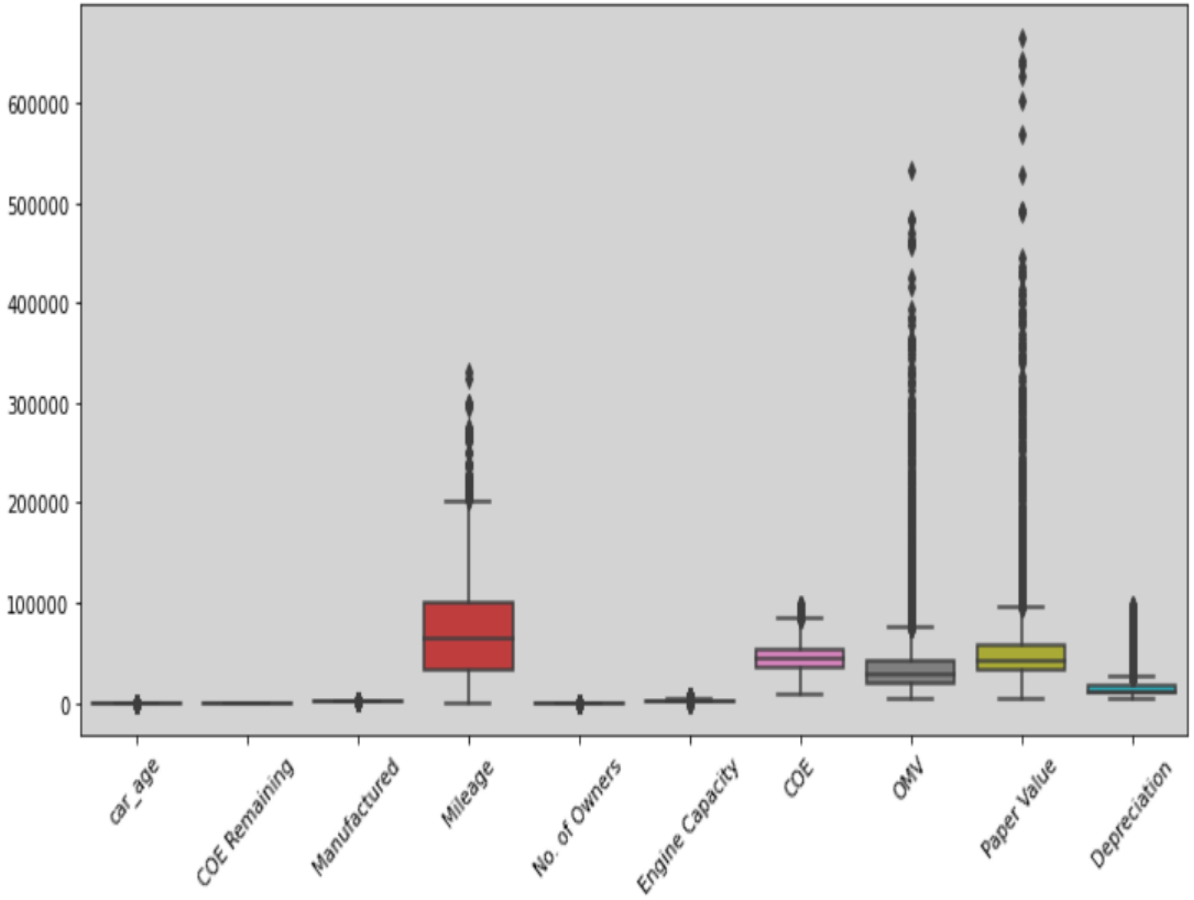
- “make”
  - We have 50 classes in this feature and we will have extra 49 columns if we use Dummy Encoding
  - We will group them in 2 groups, europe and others so that we have only 2 classes in this feature

	make
0	europe
1	others

- We will drop “model” (2193 classes) since “OMV”, “Engine Capacity” and “fuel_type” can somehow describe the car (is it exotic? is it petrol, diesel, elctric driven?)

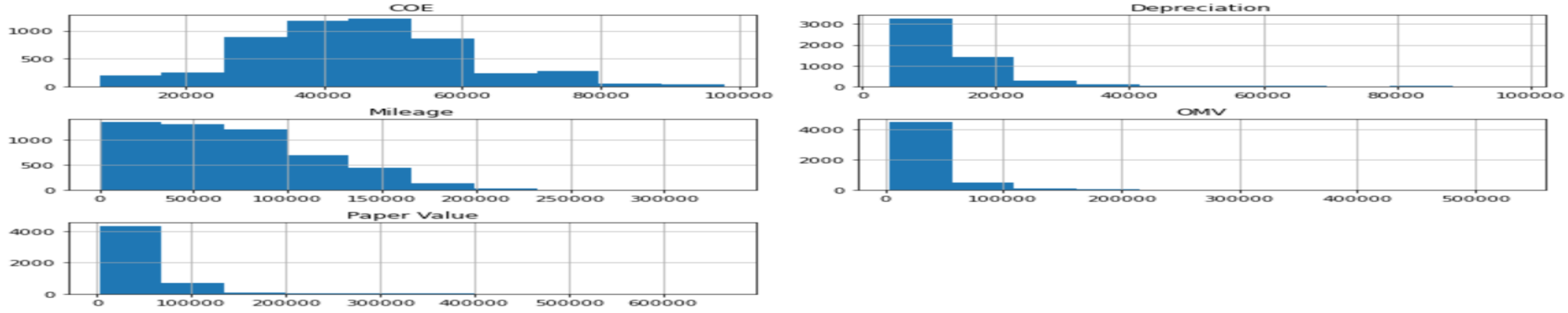
- Split the dataset into X (features) and y (label)

Data Pre-processing

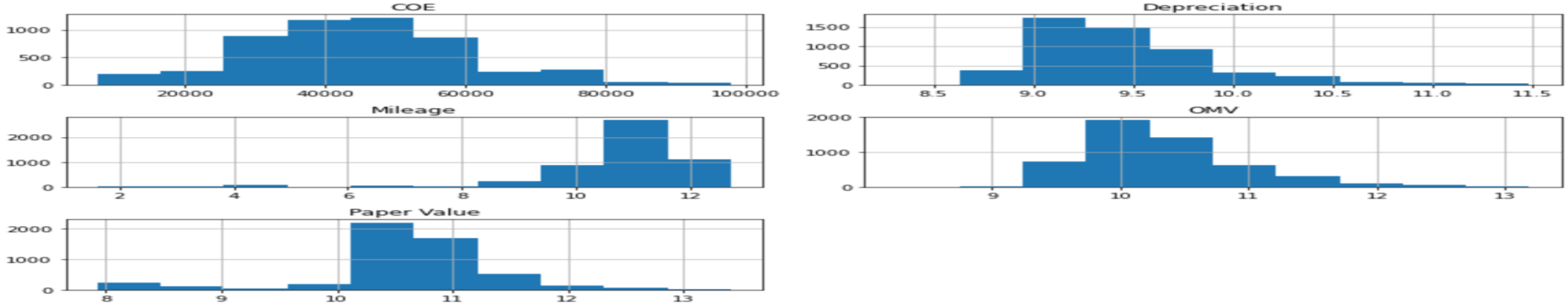


- There are a lot of outliers in “Mileage”, “COE”, “OMV”, “Paper Value” and “Depreciation”. All these outliers are related to the car usage and brands and we will keep them.

Data Pre-processing



- Other the “COE”, the rest are all right skewed
- We will adjust them using log



Data Pre-processing

- Convert categorical features to numerical features

Dummy Encoding

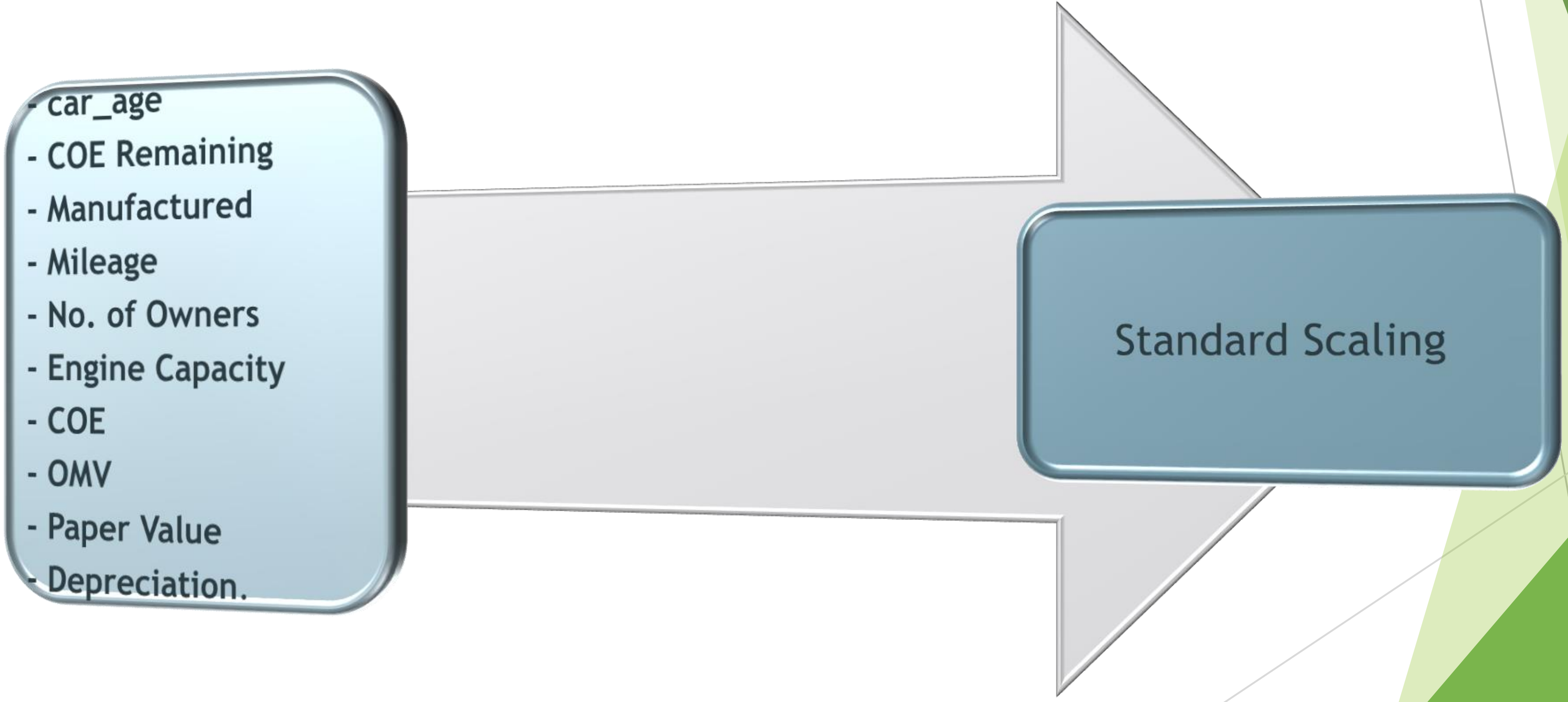
- make
- Transmission
- fuel_type,
- electric

Frequency Encoding

- Type

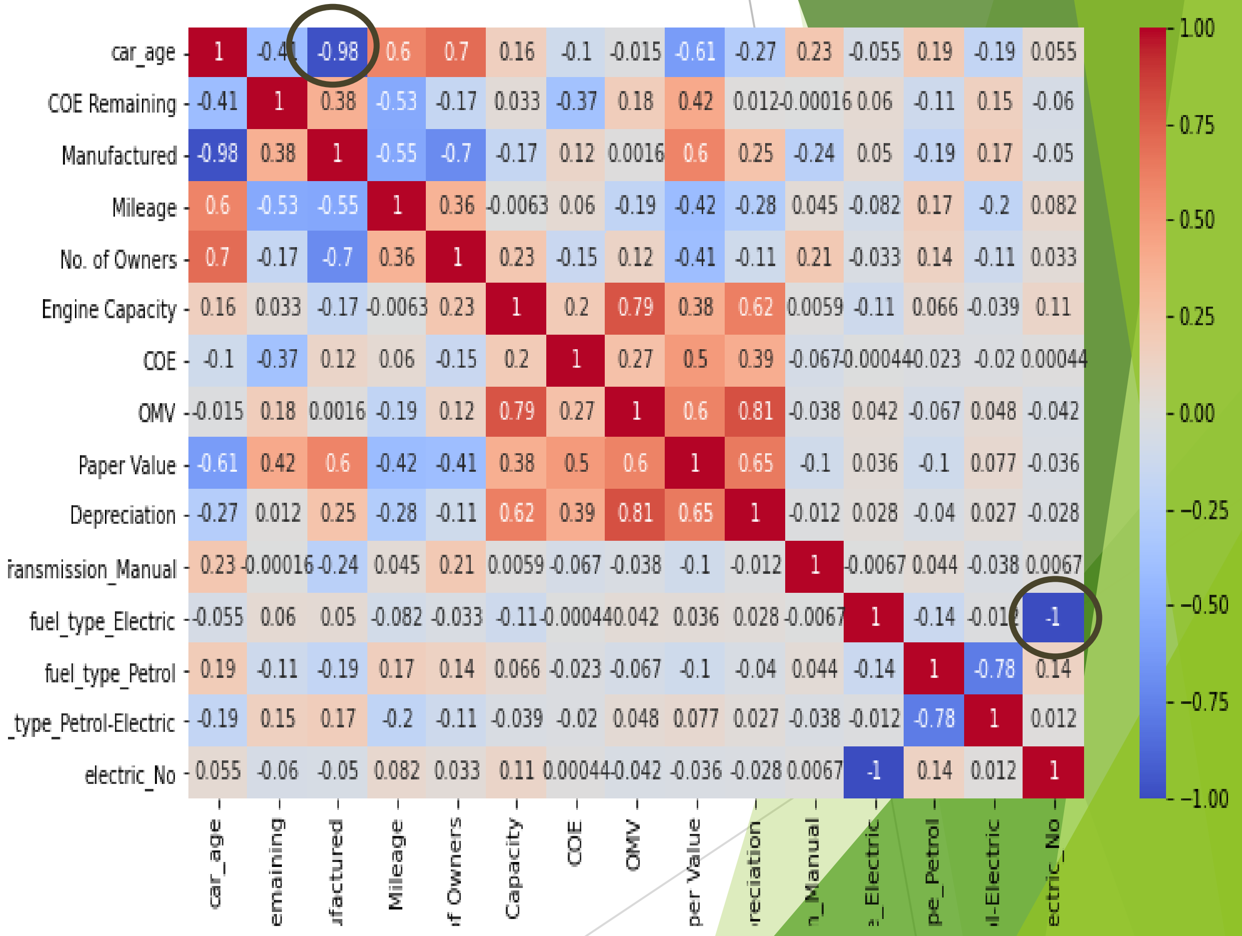
Data Pre-processing

- Split the dataset into train and test set
- Perform Feature Scaling



Data Pre-processing

- Multicollinearity check
  - Highly correlated (> 0.8 or < -0.8):
    1. “Manufactured”, “car_age”
    2. “electric_No”, “fuel_type_Electric”
  - Drop “Manufactured” and “electric_No”



Model Selection and Training

- We are using **Lazy Predict** to see which basic model work the best.
- **Extra Trees Regressor** and **Gradient Boost Regressor** are the top 2
- We will train and tune this 2 model and see the performance
- As we are keeping the outliers, we will use ***MAE** as the metric to evaluate the model since it is robust to outliers

* The average absolute difference between the actual value and predicted value

Model	Adjusted R-Squared	R-Squared	RMSE
ExtraTreesRegressor	0.98	0.98	16064.31
GradientBoostingRegressor	0.96	0.96	20437.44
RandomForestRegressor	0.96	0.96	21570.59
HistGradientBoostingRegressor	0.96	0.96	21659.62
LGBMRegressor	0.96	0.96	21956.17
BaggingRegressor	0.95	0.95	22288.95
XGBRegressor	0.95	0.95	22680.95
ExtraTreeRegressor	0.95	0.95	23856.94
KNeighborsRegressor	0.94	0.94	26042.92
DecisionTreeRegressor	0.93	0.93	27339.44
PoissonRegressor	0.92	0.92	28746.25
GaussianProcessRegressor	0.81	0.81	45467.47
AdaBoostRegressor	0.80	0.80	46523.04
GammaRegressor	0.79	0.79	47498.17
LarsCV	0.78	0.79	48126.94
LassoLarsIC	0.78	0.79	48206.95
LassoCV	0.78	0.78	48401.06
SGDRegressor	0.78	0.78	48411.37
LassoLars	0.78	0.78	48420.02
LassoLarsCV	0.78	0.78	48437.95
BayesianRidge	0.78	0.78	48443.36
Ridge	0.78	0.78	48462.16
RidgeCV	0.78	0.78	48462.16
Lasso	0.78	0.78	48471.19
Lars	0.78	0.78	48472.04
LinearRegression	0.78	0.78	48472.04
TransformedTargetRegressor	0.78	0.78	48472.04
ElasticNet	0.75	0.76	51449.71
TweedieRegressor	0.72	0.72	54937.84
GeneralizedLinearRegressor	0.72	0.72	54937.84
OrthogonalMatchingPursuitCV	0.71	0.71	55738.25
HuberRegressor	0.68	0.68	58828.42
PassiveAggressiveRegressor	0.67	0.67	59871.49
RANSACRegressor	0.61	0.62	64498.22
OrthogonalMatchingPursuit	0.56	0.57	68524.96
ElasticNetCV	0.04	0.06	101399.82
DummyRegressor	-0.02	-0.00	104449.21
NuSVR	-0.04	-0.02	105612.89
SVR	-0.10	-0.08	108411.96
KernelRidge	-0.34	-0.32	119817.11
MLPRegressor	-0.82	-0.80	139940.46
LinearSVR	-1.05	-1.02	148469.71

Model Selection and Training

Extra Tree Regressor

Train the basic model setting “criterion” to ‘mae”

Train with tuned hyperparameters:

- n_estimator : 120
- max_depth : 33
- min_sample_split : 1
- min_sample_leaf : 3
- max_features : “auto”

Gradient Boosting Regressor

Train the basic model setting “criterion” to ‘mae”

Train with tuned hyperparameters:

- Learning_rate : 0.1
- max_depth: 5
- n_estimators: 1000
- max_features : “auto”
- min_samples_leaf : 1
- min_samples_split : 6

Model Selection and Training

Metric	Extra Tree (basic)	Extra Tree (tuned)	Gradient Boosting (basic)	Gradient Boosting (tuned)
RMSE	16218.794	16334.472	21069.259	20454.419
MAE	4387.097963671	4425.405329827	7033.2142648	5209.739405823
MAPE	3.25%	3.31%	5.69%	3.36%
R2	0.975839465280	0.975493594982	0.9592274429	0.961572360203

- The basic Extra Tree Regressor will be our final model since it has the smallest average difference between the actual and predicted value.

Top 5 Overvalued and Undervalued

- Overvalued

make	model	car_age	COE Remaining	Manufactured	Mileage	No. of Owners	Transmission	Engine Capacity	fuel_type	COE	OMV	Paper Value	Depreciation	Type	electric	Price	Predict	diff
PORSCHE	PORSCHE 911 C2 COUPE (COE TILL 03/2025)	26	46	1995	113792	6	Manual	3600	Petrol	73035	108671	28254	11632	Sports Car	No	450000	104840.59	345159.41
FERRARI	FERRARI 575M MARANELLO (COE TILL 08/2024)	17	38	2004	55584	4	Auto	5748	Petrol	66834	259435	21607	12992	Sports Car	No	420000	188632.93	231367.07
MAZDA	MAZDA RX7 EFINI (COE TILL 04/2029)	28	95	1992	69000	6	Manual	1308	Petrol	26175	52183	20825	42480	Sports Car	No	338000	269112.28	68887.72
MCLAREN	MCLAREN 720S	3	86	2018	39000	1	Auto	3994	Petrol	32551	213158	290165	79860	Sports Car	No	751988	708055.64	43932.36
ROLLS-ROYCE	ROLLS-ROYCE DAWN	2	94	2016	14000	1	Auto	6592	Petrol	32909	417109	567902	11817	Sports Car	No	1288000	1251975.36	36024.64

- The top 5 overvalued cars are all sports car
- The top 3 are so called "collector items".

Top 5 Overvalued and Undervalued

- Undervalued

	make	model	car_age	COE Remaining	Manufactured	Mileage	No. of Owners	Transmission	Engine Capacity	fuel_type	COE	OMV	Paper Value	Depreciation	Type	electric	Price	Predict	diff
2162	BENTLEY	BENTLEY CONTINENTAL FLYING SPUR 4.0A V8	4	67	2015	23000	2	Auto	3993	Petrol	54901	185514	260580	57310	Luxury Sedan	No	478000	568942.16	-90942.16
1290	ROLLS-ROYCE	ROLLS-ROYCE PHANTOM (COE TILL 11/2027)	14	78	2007	66000	6	Auto	6749	Petrol	50168	461079	32754	57410	Luxury Sedan	No	375000	463541.96	-88541.96
49	LAMBORGHINI	LAMBORGHINI HURACAN LP580-2	4	68	2016	2300	2	Auto	5204	Petrol	53106	202240	282462	80600	Sports Car	No	630000	713209.64	-83209.64
1942	BMW	BMW M SERIES M8 COMPETITION CONVERTIBLE	1	112	2020	5000	1	Auto	4395	Petrol	35001	177705	251716	58990	Sports Car	No	699000	782132.04	-83132.04
757	MCLAREN	MCLAREN 650S SPIDER	5	62	2014	23000	3	Auto	3798	Petrol	57508	290157	400616	67470	Sports Car	No	598000	678814.68	-80814.68

- The top 5 undervalued cars are all european make
- 4 out of the 5 are exotic brand
- Seem worth if buyers are into exotic brand as they are undervalued by an average of around $85k (which is quite a huge sum!)

Challenges Encounter

Doing web scrapping to get the dataset

- Need to solve “Stale Element Reference Exception” error which means the element we are looking for is no longer in the DOM
- Make sure the data we scraped are correctly stored in their respective Features

Cleaning the dataset as it is a raw data

- “Error” values in the features like “Electric” in feature **Engine Capacity**
- Wrongly classifying of the car type by the advertiser

How to do encoding for feature “make” as there are 50 classes inside

- May run into “curse of dimensionality” if using Dummy Encoding
- Frequency Encoding will cause 2 or more brands with the same frequency to have the same encoded values

Conclusion

- We got a model that can predict the selling price with Mean Absolute Percentage Error (MAPE, which is MAE in percentage term) of 3.25% which we are quite satisfied

- As having a car is a costly expenses in Singapore with lots of hidden cost like insurance, road tax and maintenance. With this model, buyers can estimate the price of the car that they are interested in to decide if the car is worth the price that are advertised and work their budget from there

- Other approach that we might want to try:
  - Scrape the data after current COE premium is release
  - Add features for COE premium and COE category for each car
  - Try using Neural Network and see if the result is better

- https://github.com/andychew8015/Capston_Project-Predict_Used_Car_Selling_Price

QUESTION?