Predicting Automobile MSRP with Descriptive Traits

Michael Buck

Background

- The purpose of this project was to attempt to use data that described the traits of an automobile to predict the MSRP of the automobile
- It is possible that a model that could do this accurately could provide insight into how manufacturers set the MSRP of their vehicles.
- The data being used was obtained from Kaggle user CooperUnion who scraped the data and uploaded it to Kaggle

What does the data look like?

Not pictured: MSRP

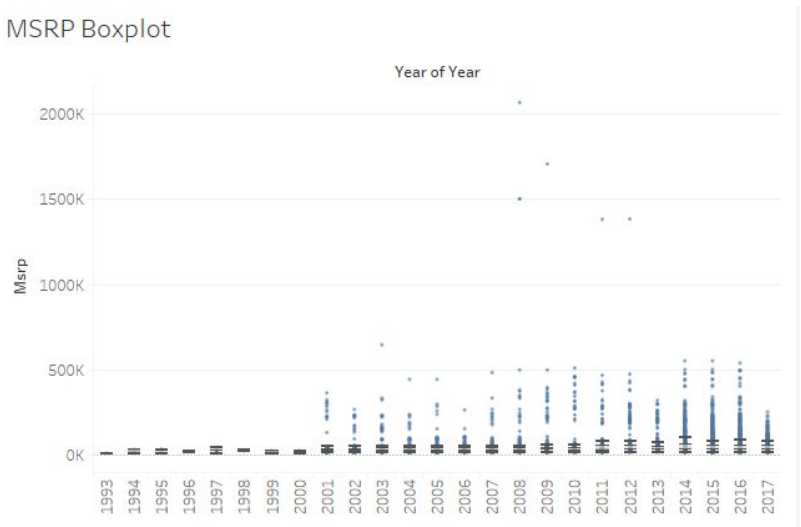
	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	highway MPG	city mpg	Popularity
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	26	19	3916
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	28	19	3916
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	28	20	3916
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	28	18	3916
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	28	18	3916

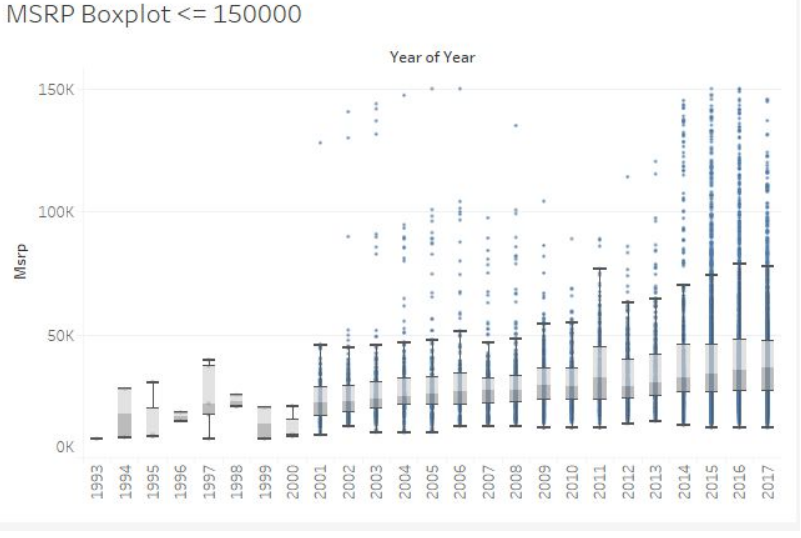
Data Cleaning:

- Before going into EDA, I'm going to make a few quick notes on the state of the data as I obtained it.
  - The MSRP data for many of the vehicles in the early years of this dataset was wrong, and as such a large portion of the data had to be dropped as it seemed to have filler values for older vehicles($2000 typically). The final data had a bit more than 10000 vehicles in it.
  - Beyond that the majority of the data was fairly clean, besides a couple values that were easy to look up and fix.

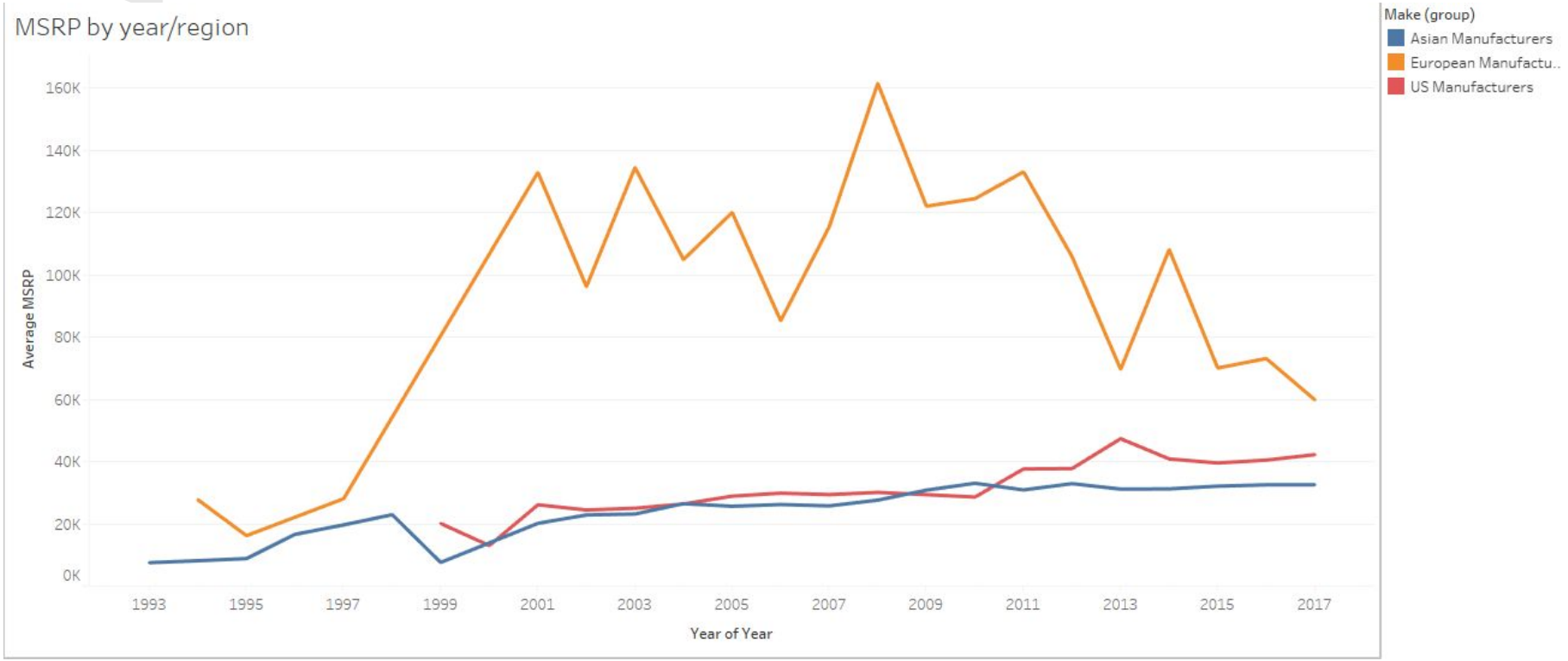
MSRP data

Observing the MSRP data here, we see that as time progresses the IQR of prices seems to increase. We can also observer that the data contains more raw observations the later we go in time. This is largely a result of having to drop most of the early data from the dataset, but the original data did have fewer cars in the earlier years. The second plot was made to help with viewing the box features, as it ignores the more extreme prices in the dataset.

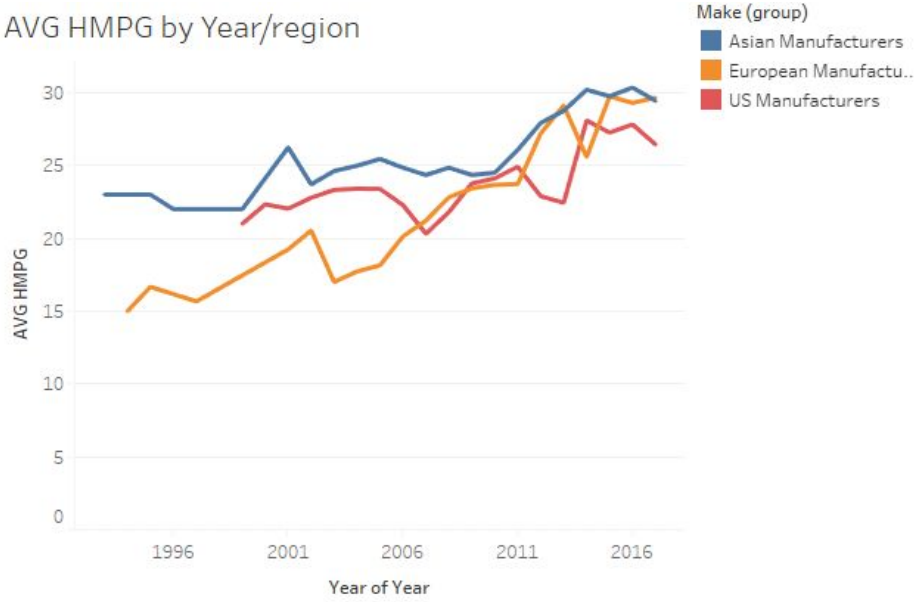


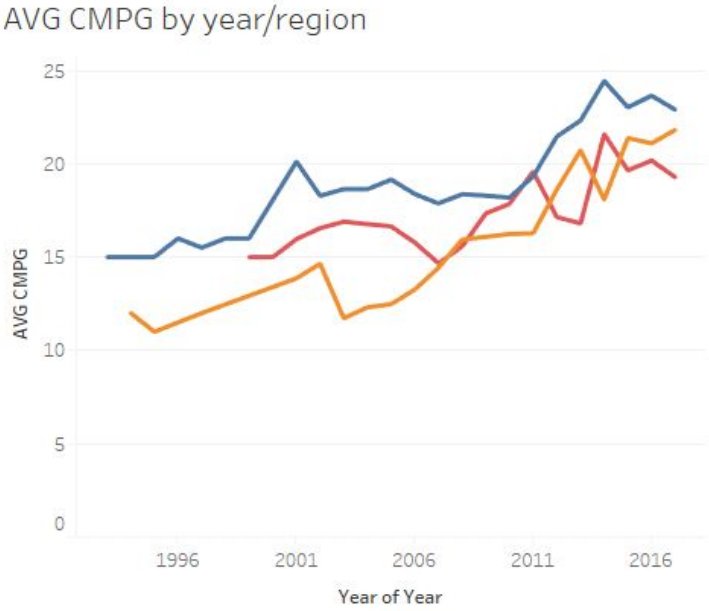


Average MSRP by Year and Region

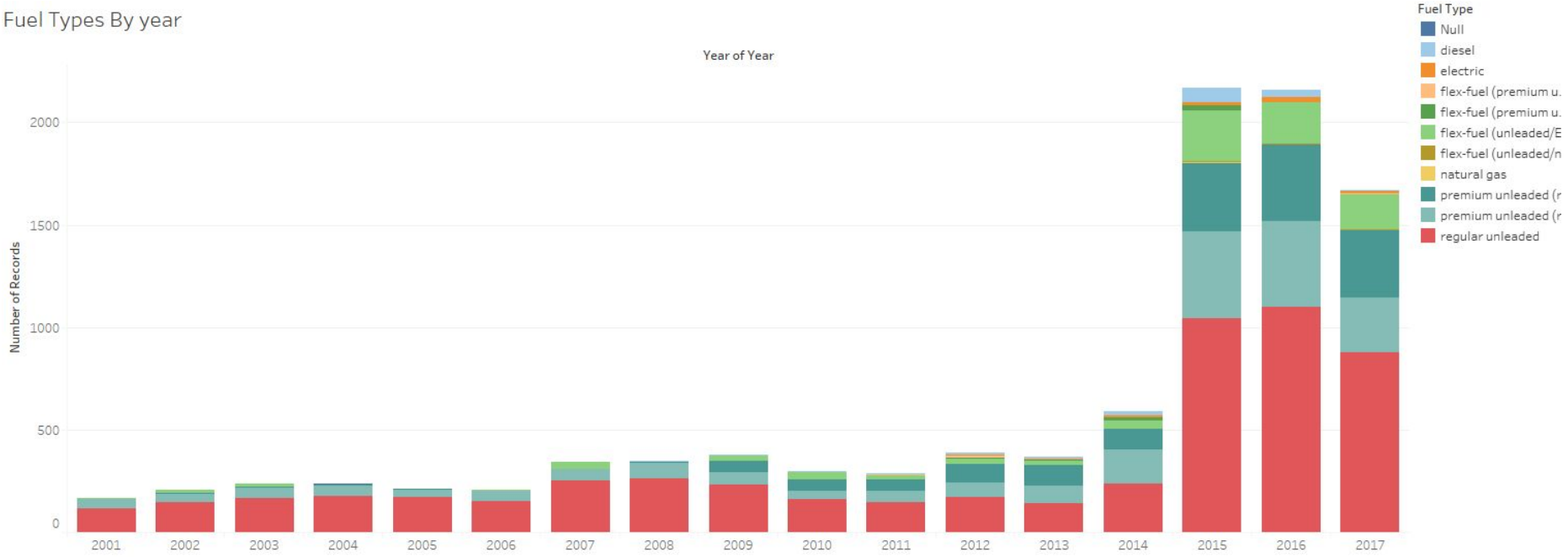


Average Highway and City Gas Mileage by Region



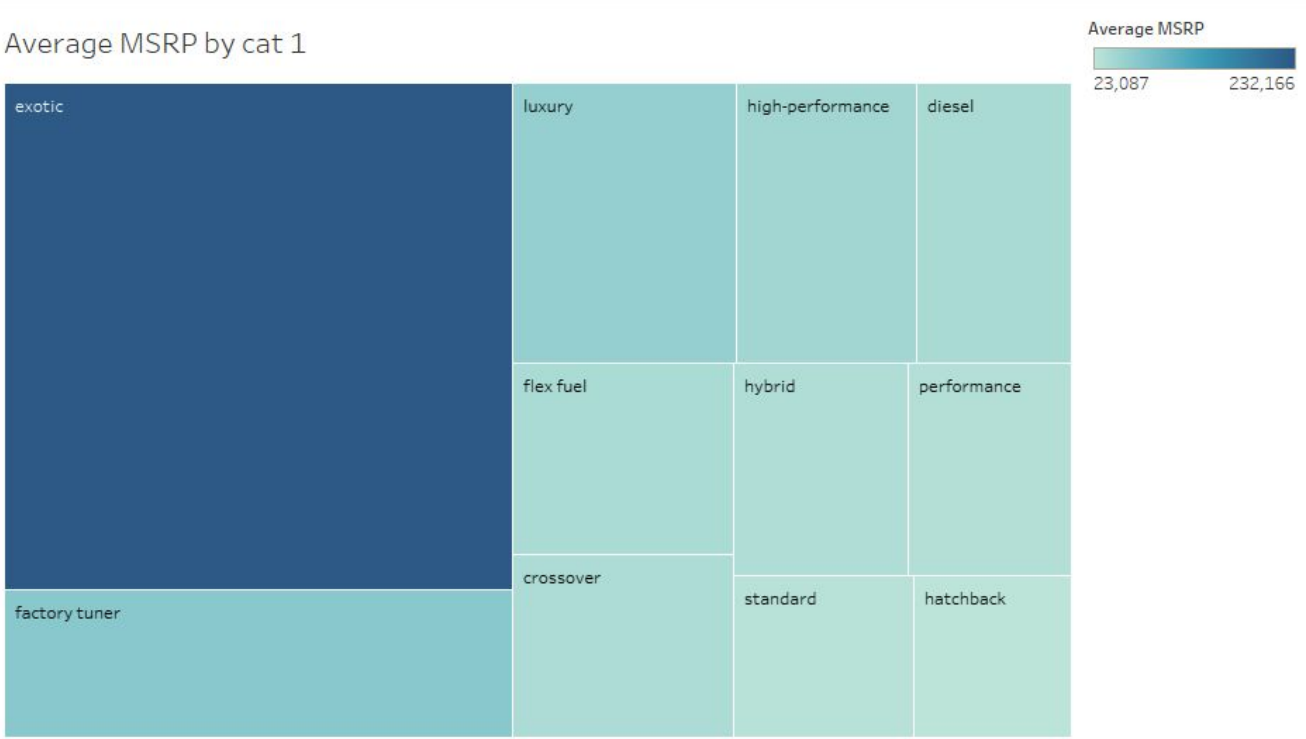


Vehicle Fuel Types By Year



This plot shows average MSRP by style designation and year.

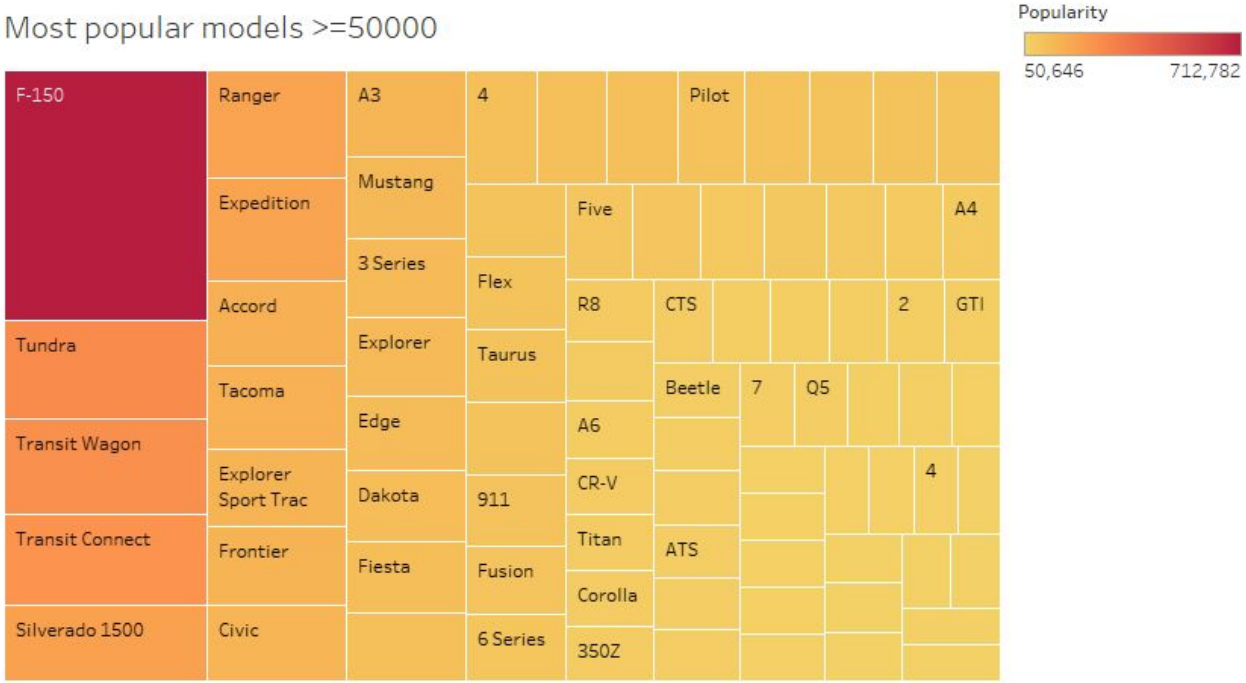
Average MSRP by Category



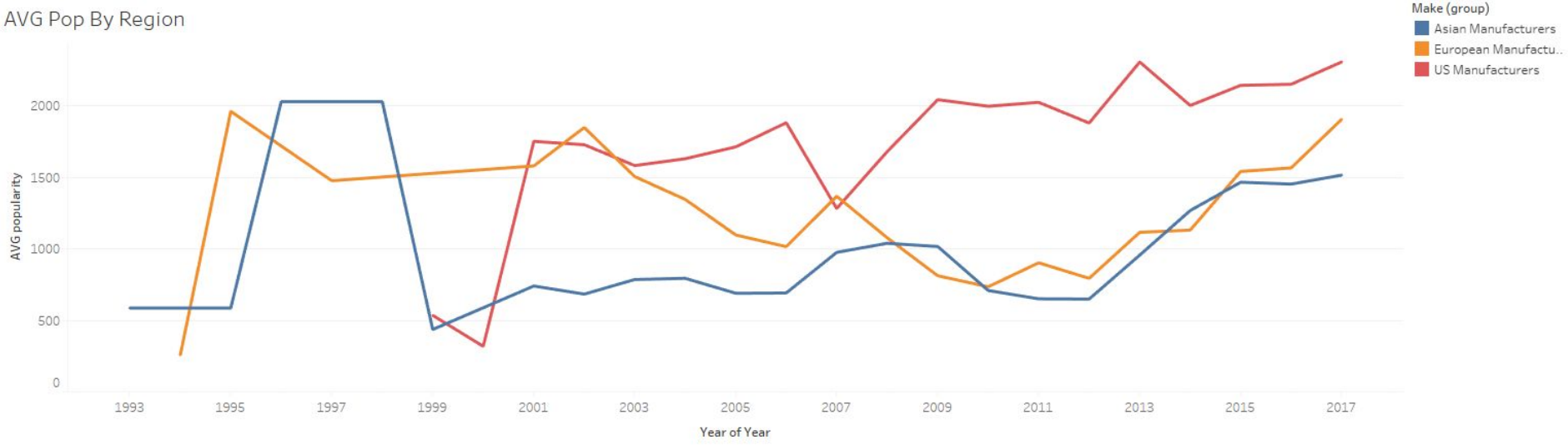
Exploratory Data Analysis & Popularity

- The previous slides were some of the visualizations I made exploring the data and a more in depth analysis of those visualizations can be found in my paper.
- The next visualizations are based off of the popularity column in the dataset, which is simply a number, that was somehow scraped from twitter, but it is never explained in the data documentation where the number came from.
- My assumption is the number is a pure count of tweets, however since the user who scraped the data has never clarified how the data was obtained, that assumption may be wrong.

Sum of popularity values by model



Average Popularity by Region



Model Building

- After extensive EDA, I began to prepare for constructing a model. The plan was to use an XGBoost regressor, and as such I needed to perform some more cleaning on the data to get it model-ready.
- Categorical features of the data were one hot encoded.
- The category variable had up to five categories in one vehicle, however the vast majority of vehicles had one category, and many had two categories, so that was the number of categories retained before one hot encoding.

Model Building

- Some Horsepower values were missing for vehicles, and for the sack of time were imputed with the mean value as it seemed reasonable to do.

- Bugatti vehicles were removed from the dataset before modeling due to their prices being extreme outliers, and there were only three in the dataset to begin with.

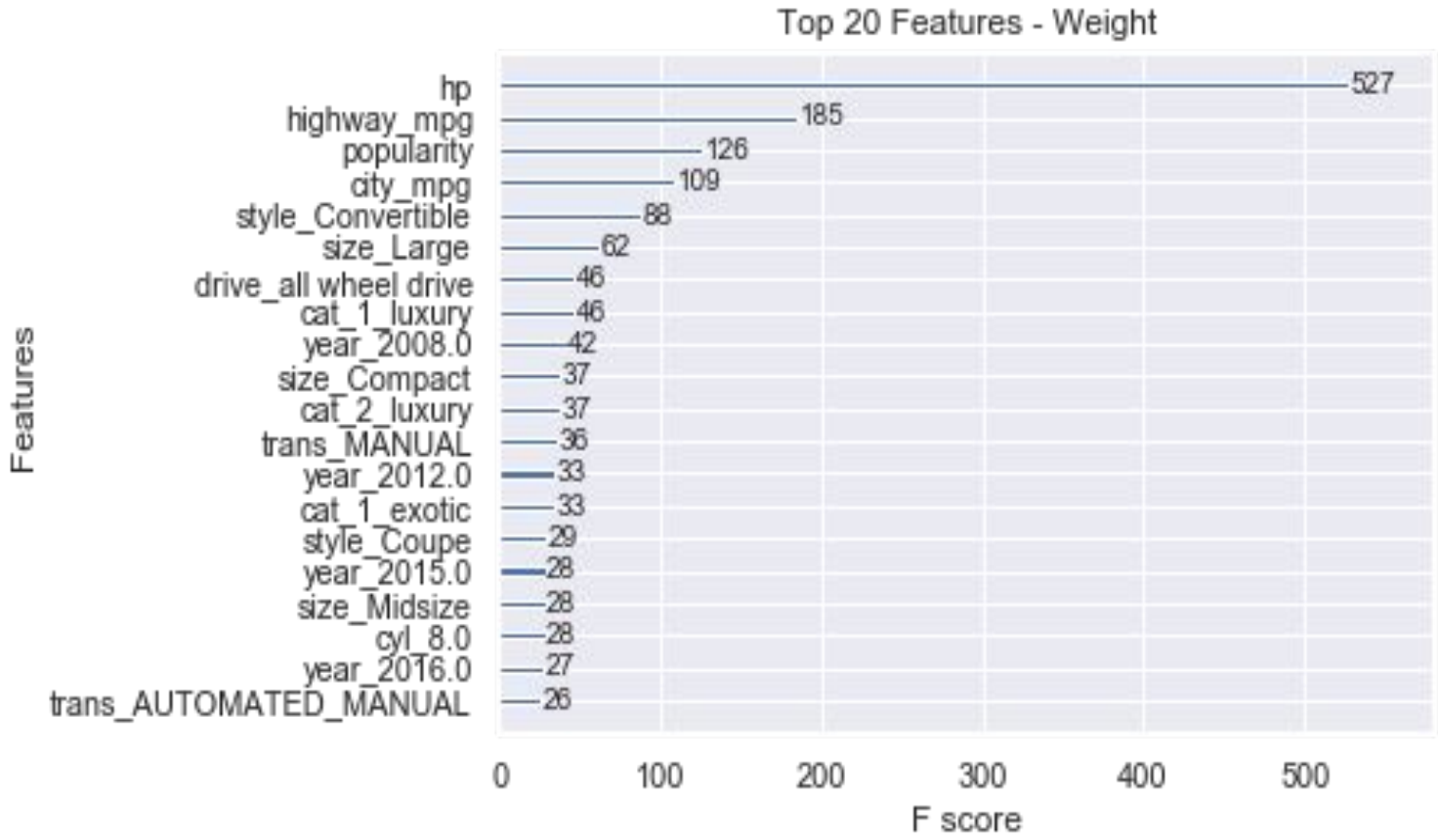
Model Building

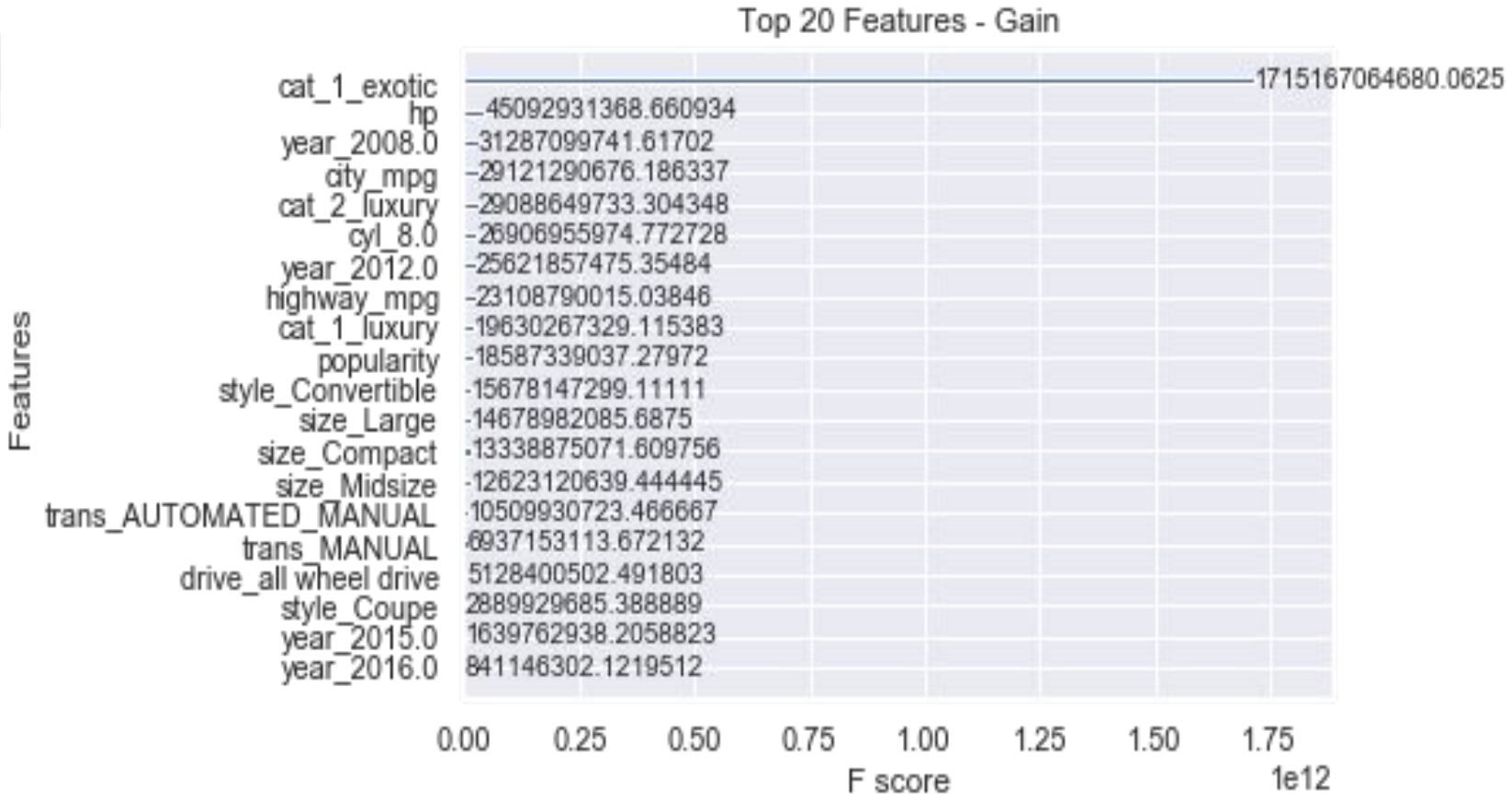
- The out of the box XGBoost regressor had an RMSE of around $14000. After looking at the results of the out of the box model, I discussed the results with my mentor and we decided to do a bit more feature engineering.
- The top 30 most common makes and models of vehicles were retained as categories but everything else was simply labelled ‘other’ in its respective make or model category.
- Numerical variables were standardized in order to put them into a better scale for the model.

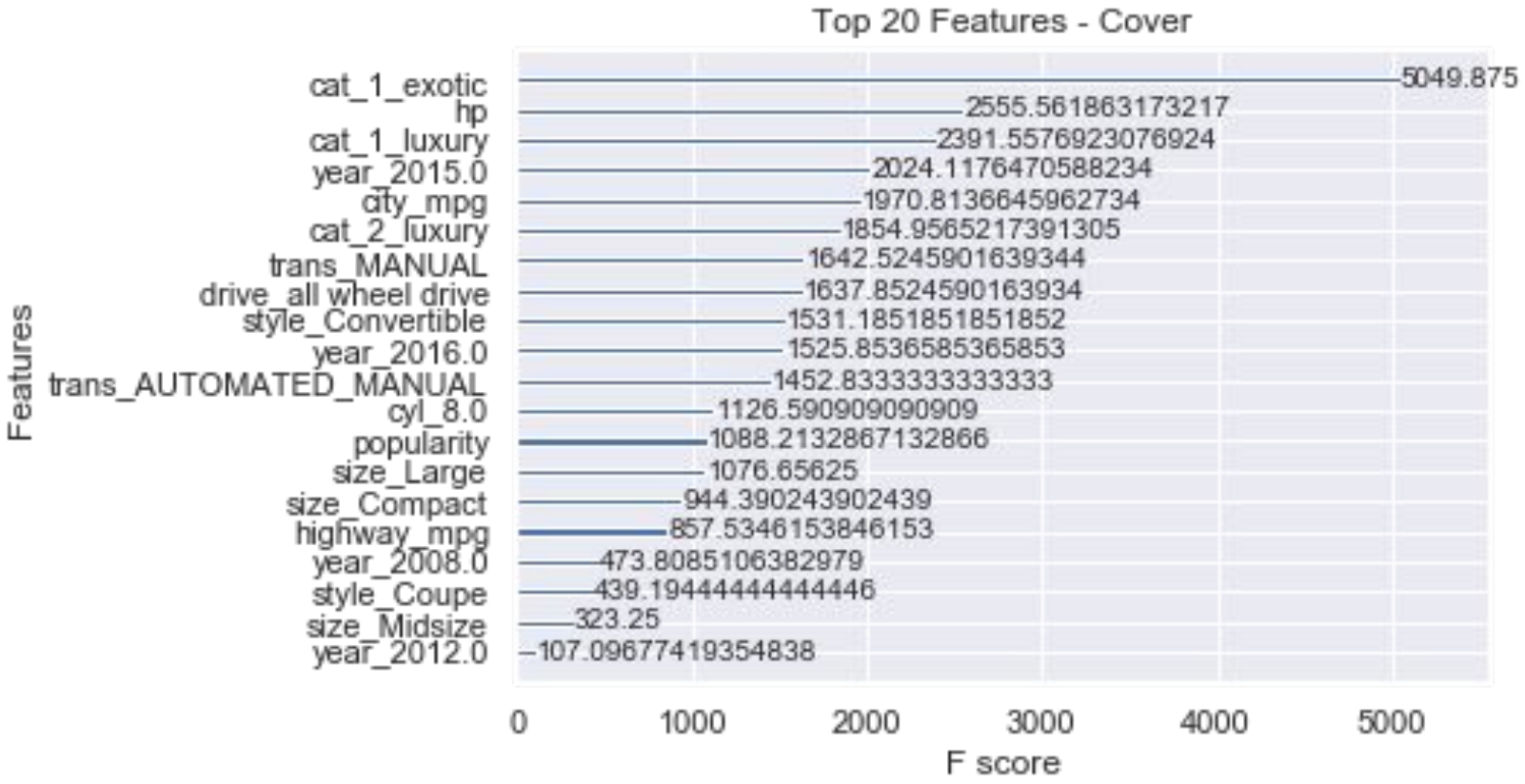
Model Building

- After re-training the model on the new features and performing three-fold cross validation to tune the XGBoost parameters, the new model had an RMSE around $12000.

- This was a good reduction in error beyond the out of the box model, so I then looked to the feature importances of this model to see what information I could glean from them.


Top 20 Features - Weight
Features
hp
highway_mpg
popularity
city_mpg
style_Convertible
size_Large
drive_all wheel drive
cat_1_luxury
year_2008.0
size_Compact
cat_2_luxury
trans_MANUAL
year_2012.0
cat_1_exotic
style_Coupe
year_2015.0
size_Midsize
cyl_8.0
year_2016.0
trans_AUTOMATED_MANUAL
527
185
126
109
88
62
46
46
42
37
37
36
33
33
29
28
28
28
27
26
0
100
200
300
400
500
F score


Top 20 Features - Gain
Features
cat_1_exotic
hp
year_2008.0
city_mpg
cat_2_luxury
cyl_8.0
year_2012.0
highway_mpg
cat_1_luxury
popularity
style_Convertible
size_Large
size_Compact
size_Midsize
trans_AUTOMATED_MANUAL
trans_MANUAL
drive_all wheel drive
style_Coupe
year_2015.0
year_2016.0
1715167064680.0625
45092931368.660934
31287099741.61702
29121290676.186337
29088649733.304348
26906955974.772728
25621857475.35484
23108790015.03846
19630267329.115383
18587339037.27972
15678147299.11111
14678982085.6875
13338875071.609756
12623120639.444445
10509930723.466667
6937153113.672132
5128400502.491803
2889929685.388889
1639762938.2058823
841146302.1219512
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
F score
1e12


Top 20 Features - Cover
Features
cat_1_exotic
hp
cat_1_luxury
year_2015.0
city_mpg
cat_2_luxury
trans_MANUAL
drive_all wheel drive
style_Convertible
year_2016.0
trans_AUTOMATED_MANUAL
cyl_8.0
popularity
size_Large
size_Compact
highway_mpg
year_2008.0
style_Coupe
size_Midsize
year_2012.0
5049.875
2555.561863173217
2391.5576923076924
2024.1176470588234
1970.8136645962734
1854.9565217391305
1642.5245901639344
1637.8524590163934
1531.1851851851852
1525.8536585365853
1452.8333333333333
1126.590909090909
1088.2132867132866
1076.65625
944.390243902439
857.5346153846153
473.8085106382979
439.19444444444446
323.25
107.09677419354838
0
1000
2000
3000
4000
5000
F score

XGBoost feature importances

- XGBoost's feature importances can be described as follows:
  - 'Weight' is the number of times a feature appears in a tree
  - 'Gain' is the average gain of splits which use the feature
  - 'Cover' is the average coverage of splits which use the feature where coverage is defined as the number of samples affected by the split
- I wrote a function to run the model again using the top 20 features from these three types of feature importances to see if I could find a better performing model using a smaller number of features(results on next slide)

RMSE of top n features by feature importance type

```
weight
```

```
['N features: 5 Training RMSE: 10421.695191427254 Testing RMSE: 13459.171635771176',
 'N features: 10 Training RMSE: 9297.914542769116 Testing RMSE: 12050.632426005457',
 'N features: 15 Training RMSE: 7702.426289889635 Testing RMSE: 13234.070034514787',
 'N features: 20 Training RMSE: 7748.7061473725635 Testing RMSE: 13476.853568811019']
```

```
gain
```

```
['N features: 5 Training RMSE: 14620.592552624314 Testing RMSE: 21534.03438222079',
 'N features: 10 Training RMSE: 11945.29749023505 Testing RMSE: 21368.057010638182',
 'N features: 15 Training RMSE: 7942.67971610978 Testing RMSE: 13076.601145760527',
 'N features: 20 Training RMSE: 7819.9329234414545 Testing RMSE: 13938.995996005846']
```

```
cover
```

```
['N features: 5 Training RMSE: 14729.632607286805 Testing RMSE: 20636.954277761943',
 'N features: 10 Training RMSE: 9274.243284892202 Testing RMSE: 12233.163047120297',
 'N features: 15 Training RMSE: 8055.573687133821 Testing RMSE: 11164.69377196283',
 'N features: 20 Training RMSE: 7684.710130756485 Testing RMSE: 13859.971583039205']
```

Using Feature Importance to Select Features

- Both cover and weight had RMSEs that were lower than the model that used all of the features, so I checked each of those on a more granular level and found the cover features produced the smallest testing RMSE at 16 features.

```
'N features: 14 Training RMSE: 8347.090825297635 Testing RMSE: 10877.2445485881',
'N features: 15 Training RMSE: 8055.573687133821 Testing RMSE: 11164.69377196283',
'N features: 16 Training RMSE: 8225.7454188634 Testing RMSE: 10194.862246313296',
```

The ‘best’ features

- The best features found to reduce RMSE using this process are named in the image on this slide.
- Labels that identify a car as ‘expensive’ like ‘luxury’ or ‘exotic’ as well as ‘convertible’ all seem to be important.
- 2015 and 2016 were the most populated years in the dataset
- All of the numeric variables are included in these importances
- All of the size information is also in this model. ‘Midsize’ is not directly included however anything not large or compact was midsize.

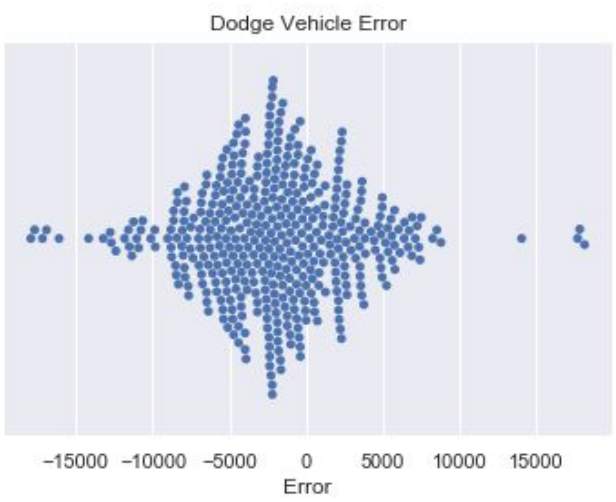
```
cover_feats[:16]
```

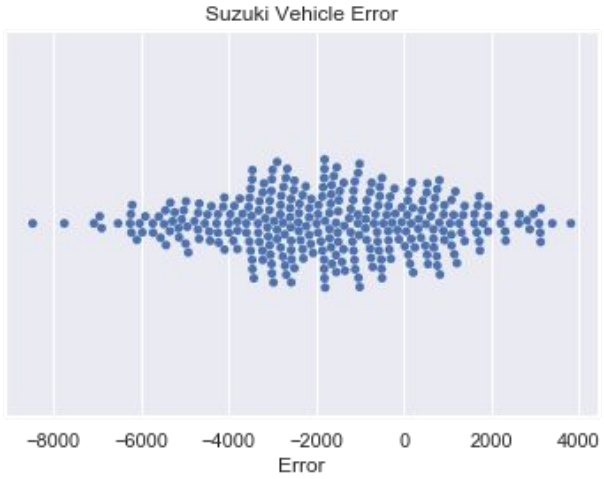
```
['cat_1_exotic',
 'hp',
 'cat_1_luxury',
 'year_2015.0',
 'city_mpg',
 'cat_2_luxury',
 'trans_MANUAL',
 'drive_all wheel drive',
 'style_Convertible',
 'year_2016.0',
 'trans_AUTOMATED_MANUAL',
 'cyl_8.0',
 'popularity',
 'size_Large',
 'size_Compact',
 'highway_mpg']
```

Error Analysis

Error Analysis

- Beyond a very small number of outliers the errors of the model appear to be normally distributed.
- The errors of the most common makes are also normally distributed around 0 beyond a few exceptions which can be seen below.





Final Thoughts

- The model seems to be relatively successful for a first attempt using a relatively limited feature set to predict something like this, however I don't think a model with an RMSE around $10,000 can be said to predict vehicle price 'accurately' as that can often be around 15% off for common vehicles. The model is better at predicting the most common makes in the dataset, so perhaps it could be more useful there.
- Were I too improve this model, I would look into creating an ensemble model that incorporates more techniques to improve the accuracy of the model, and perhaps include a few more features in the dataset such as region.