Capstone 2: Final Report

Introduction:

Vehicles are an important part of many people's lives. Whether the vehicle is for transporting ones family, commuting to work, or a vehicle for work, a vehicle is an important investment that is practically mandatory in many parts of America. With that being said, I was interested in analyzing some type of vehicular data for one of my capstone projects, and fortunately for this second capstone project I was able to do so. Using data gathered by Kaggle user CooperUnion, I wanted to analyze factors that could potentially affect the MSRP that a manufacturer chooses for a vehicle. Many factors could affect MSRP, from manufacturer biases about how they perceive the reputation of their products to cutting edge technology that is expensive to produce and sell. This specific dataset describes many of the most common features of vehicles that are evaluated by a customer while shopping for a vehicle so I believed it to be a good data set for a preliminary assessment on what may affect some manufacturers choices when suggesting a price for their vehicles. In this document I will be describing how I cleaned this data for analysis and exploratory data analysis that I performed with this data before making any predictions with models.

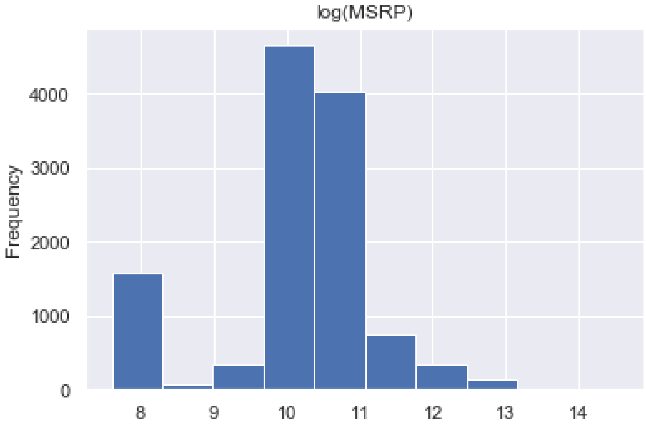
Data Description and Cleaning:

The data source for this project is linked previously, but I will take some time to describe the data and the features contained within it. The data originally contained around 12000 rows with 16 columns. The columns are:

- Vehicle identification columns:
  - Make, model, year.
- Vehicle description columns:
  - Fuel type, horsepower, number of cylinders, transmission type, drive type, number of doors, market category, vehicle size, vehicle style, highway mpg, city mpg
- A column titled "popularity" which seems to be a count of number of tweets about the vehicle in question(this feature isn't documented well in the data description).
- A column with the MSRP of each vehicle.

This dataset originally appeared to be very clean, so the cleaning process was fairly standard, at least initially. The first thing I did was to rename the columns to be shorter, more standardized, and a bit more descriptive of what they actually contained in order to ease using the data while programming. After changing the names, I took the time to look closely at each feature to try to find incorrect or missing data. The make, model, and year columns did not seem to contain any erroneous data, as all of the manufacturers were spelled correctly and I was unable to find any text errors in the model column. The other features of the data such as number of cylinders and number of doors, were similarly clean. The first hiccup in the data was in the highway MPG column, which had a single anomalously high datapoint which I was able to look up and fix.

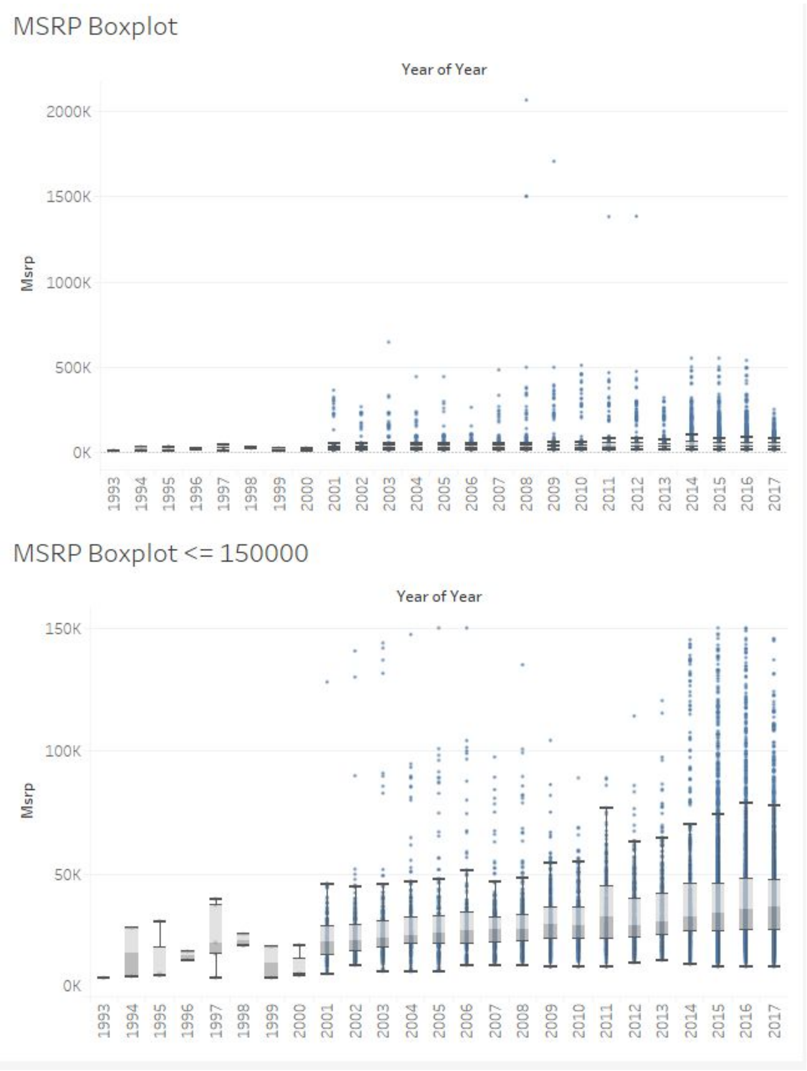
After fixing the MPG data, I went about putting the category column into a more usable form. The category column was originally a list of words that can be descriptive of vehicles such as “hatchback” or “exotic”. I went about splitting this column into multiple columns each column containing a single descriptive word from the category column. Some vehicles had up to five words in the category column. After looking at the words in these new column after splitting I came to two conclusions. The first conclusion was that for vehicles that were missing a value in the category column; they were missing a value because they did not fit any of the special niches that were described by the categorical words that were in the dataset. Since this was the case, I decided to replace the missing values in the first column with the word “Standard” since the vehicles that were missing this value simply didn’t fall into any of the categorical words initially in the data set, and did not seem to have any special defining features beyond being the standard versions of those vehicles.

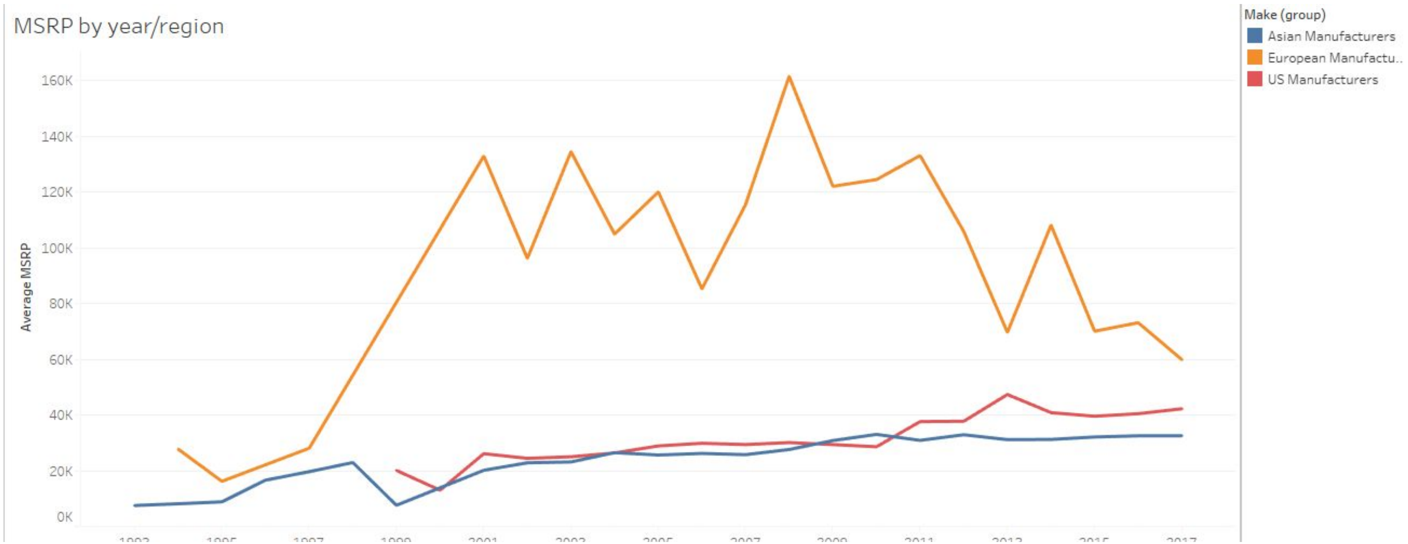


As I was looking into the numerical features of the data in more depth, I discovered something very strange occuring in the distribution of the MSRP values of the dataset, which can be observed in this histogram. The MSRP data was exponentially distributed, so I looked at the distribution of the log values of the data and found that there was an interestingly high number of relatively small values in the dataset. When I looked into these small values, they all seemed to be coming from the earliest years of the dataset, and none of them were simply off by a 0. There seemed to be a default value of 2000 that was placed into the data for cars that may not have had an MSRP when the data was scraped. There is no way to confirm why the data is erroneous but it seems that most of the data from before the year 2000 had some error that resulted in a very low MSRP being put in the place of the actual MSRP of the vehicles. Not wanting to filter out meaningful data but also not wanting to have junk data with a bad target variable to build my model with, I decided to try filtering the data by removing all of the vehicles with an MSRP less than $7,500 from my data to have more accurate MSRP values. This resulted in most of the data from before the year 2000 being dropped from the dataset, but the dataset still contained 10,278 rows of data.

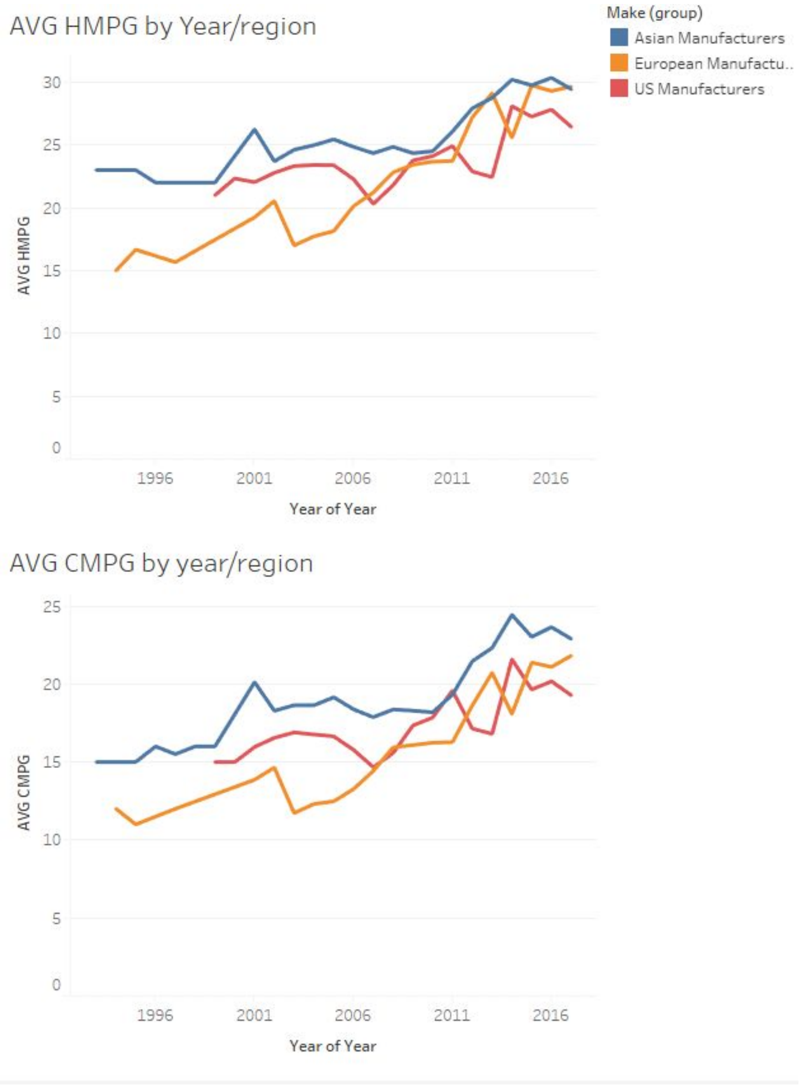
Exploratory Data Analysis:

The primary objective for the EDA of this project is to attempt to find relationships between MSRP and the other features of the dataset. As can be observed by the boxplots on the right, it seems that MSRP has slowly increased over time. With median values of MSRP not varying too significantly while slowly increasing. An interesting quirk of the MSRP values in this dataset is the ranges between years can vary wildly depending on whether or not an incredibly expensive exotic sports car was manufactured during that year. Another quirk of the data to make note of is the limited number of data remaining from the year 2000 and earlier, which was addressed earlier. Many of the visualizations moving forward may appear a bit skewed because of this fact. Below you can see a graph that plots MSRP over time by year and split by region. This specific region set was chosen because it is popular to distinguish automobile



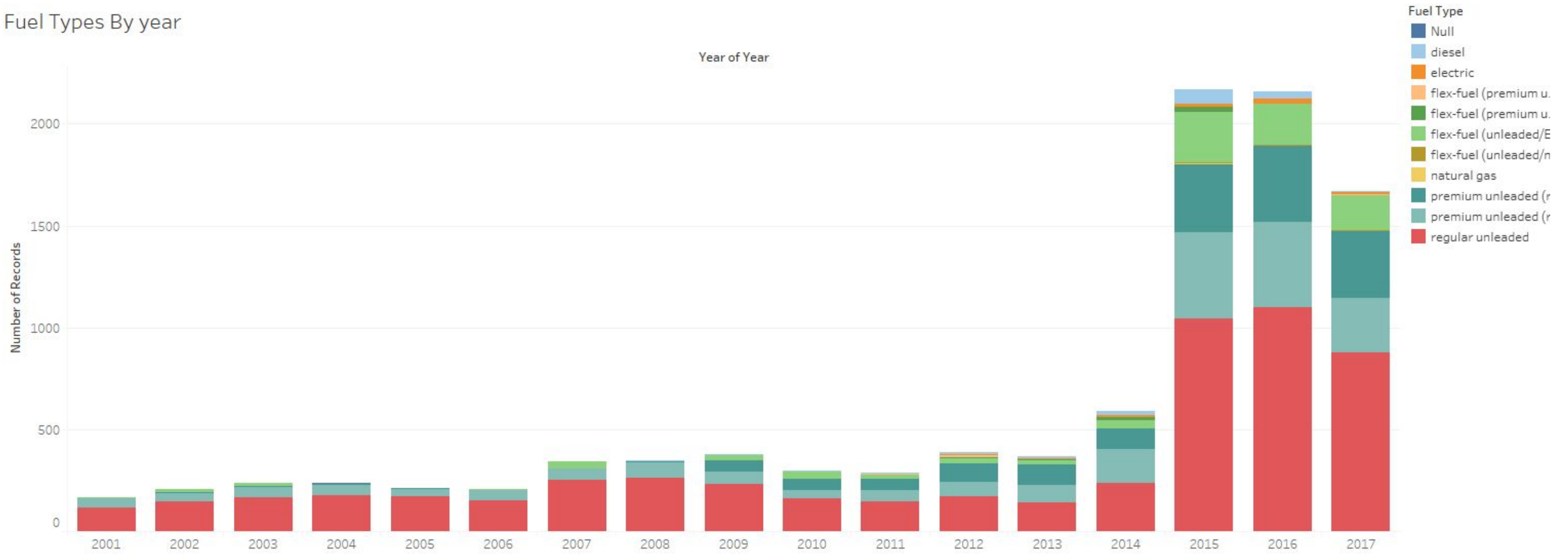


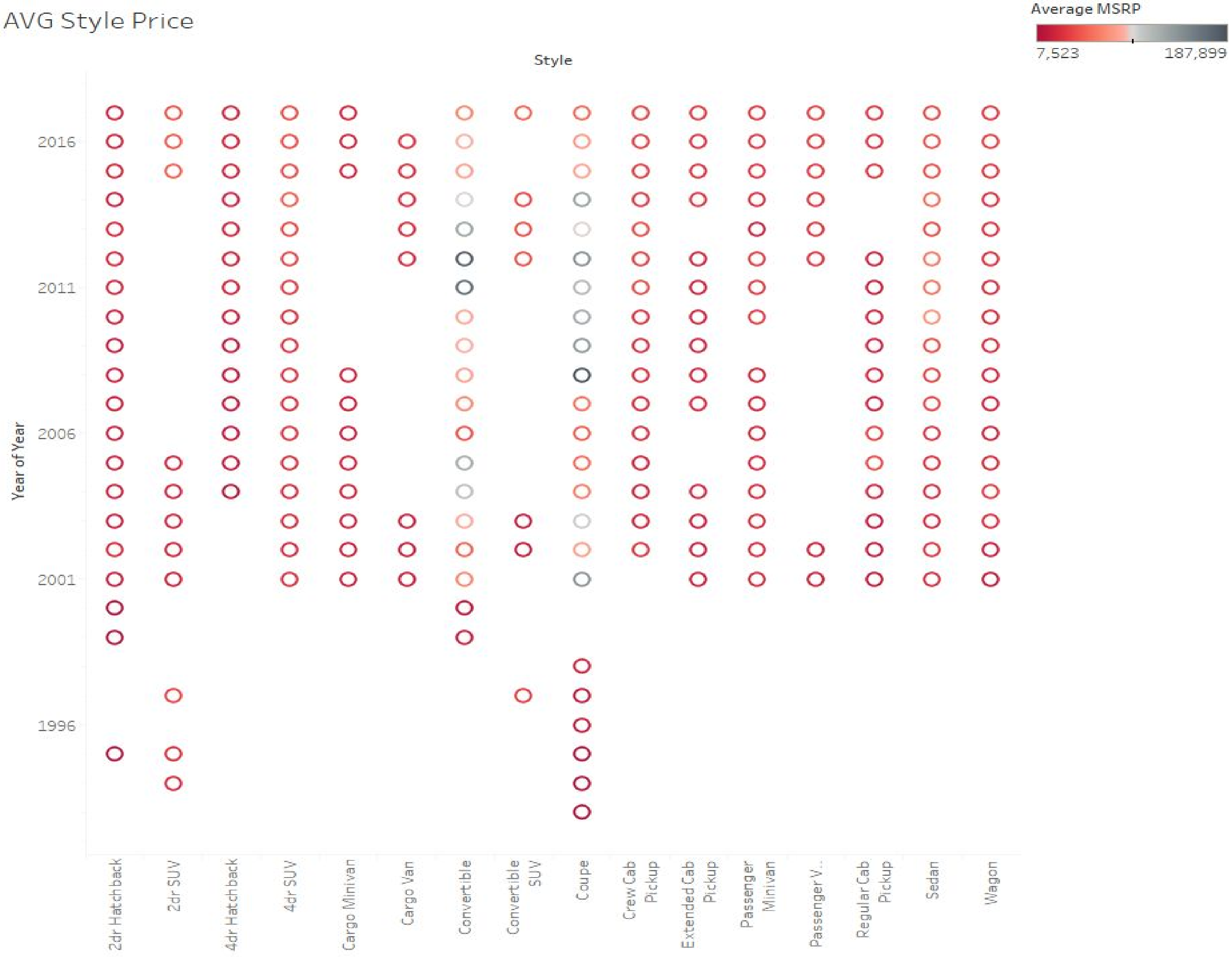
manufacturers by their general location as an American, European, or Asian automobile manufacturer. This specific plot shows average MSRP by region over time. The plot shows that average MSRP increases relatively slowly over time for both US and Asian auto manufacturers, while the average MSRP for European manufacturers is higher than the other two regions consistently, and varies wildly across years. The European average is so much higher for one primary reason; which is the fact that most ‘Exotic’ or ‘Luxury’ manufacturers, such as Ferrari, Lamborghini, and Bentley, are classified as European manufacturers. Manufacturers in the US and Asia have much more of a focus on producing economy cars or utility vehicles that do not carry a hefty price tag that something like a Bentley or Ferrari has.



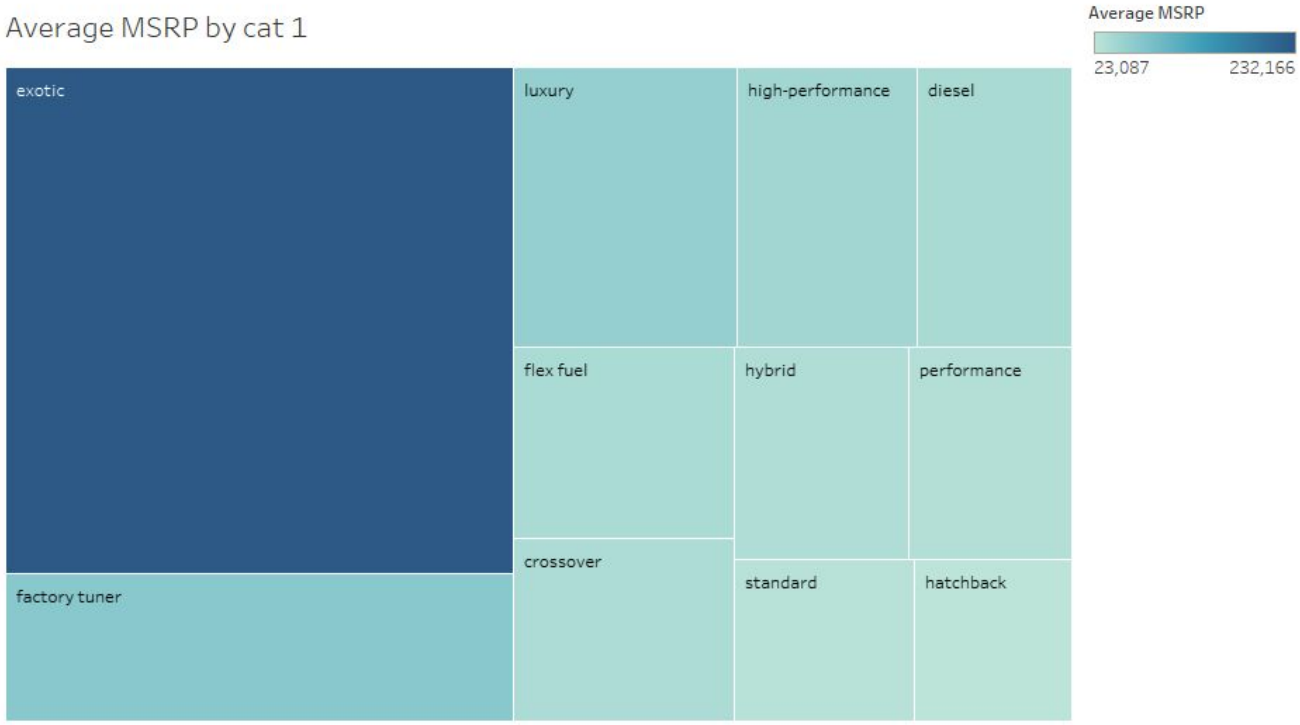
Taking a look at some of the other features of the dataset, it can be seen that each region has made some progress in increasing average city and highway fuel efficiency of their vehicles over time, with Asian manufacturers leading the way and US manufacturers lagging slightly behind the other two regions. As time has gone on, different types of fuel have been developed for vehicles such as E85, and the number of records by fuel type can be observed in the chart on the next page. It seems prudent to note that the vast majority of vehicles being manufactured today still seem to require ‘normal’ gasoline, or the premium unleaded equivalent, but proportionally there are many more flex-fuel and E85 vehicles in the 2015-2017 range than in the past in this data.

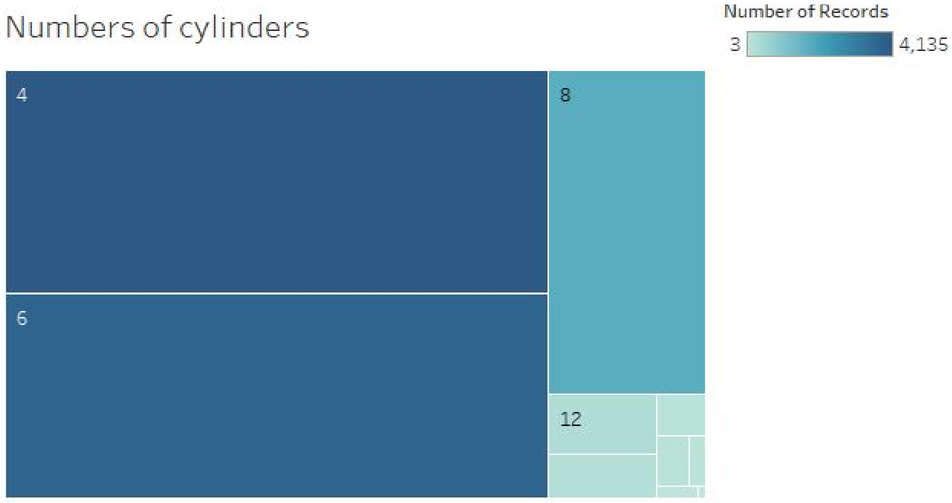
The ‘style’ a vehicle has the potential to be one of the more important aspects in regards to predicting MSRP for a vehicle. On the next page you can see a chart that shows relative average MSRP by year separated by the styles that are included in the data set. The chart reveals that most styles’ average MSRP’s do not vary much over time, and average MSRP does not seem to vary significantly between





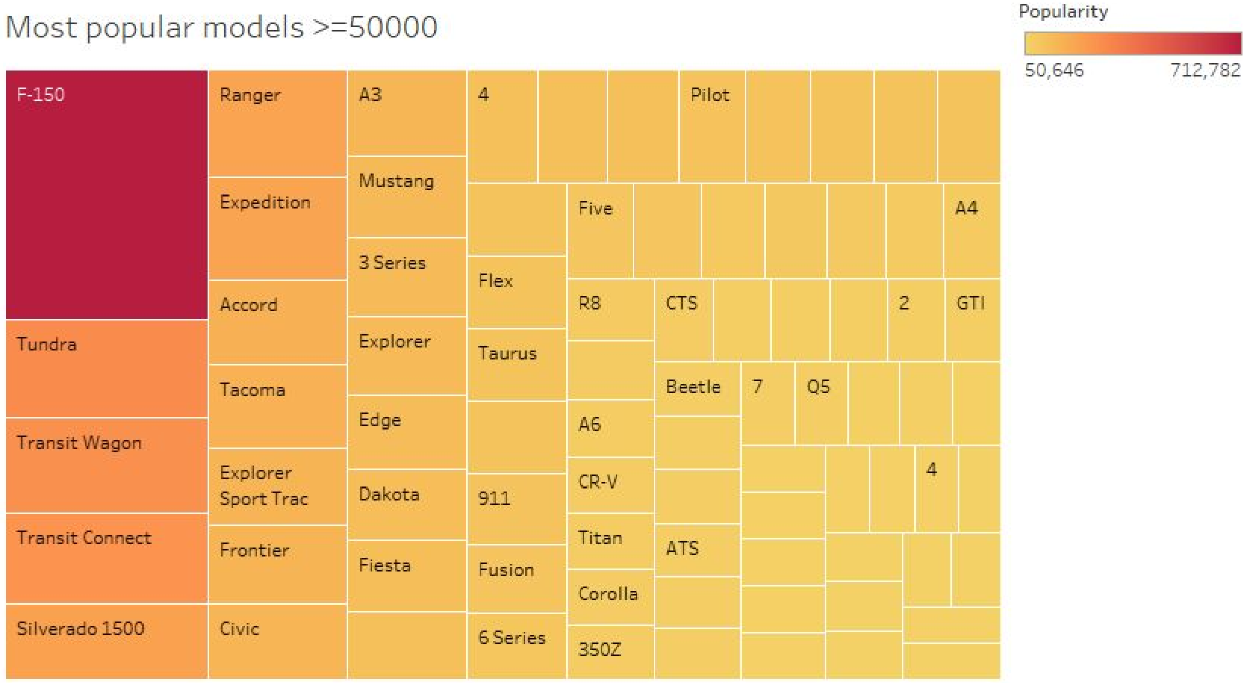
styles, with the two exceptions to this being a few years in the ‘convertible’ and ‘coupe’ columns. It does seem that those two styles are the most expensive of the columns in terms of average MSRP.


Average MSRP by cat 1

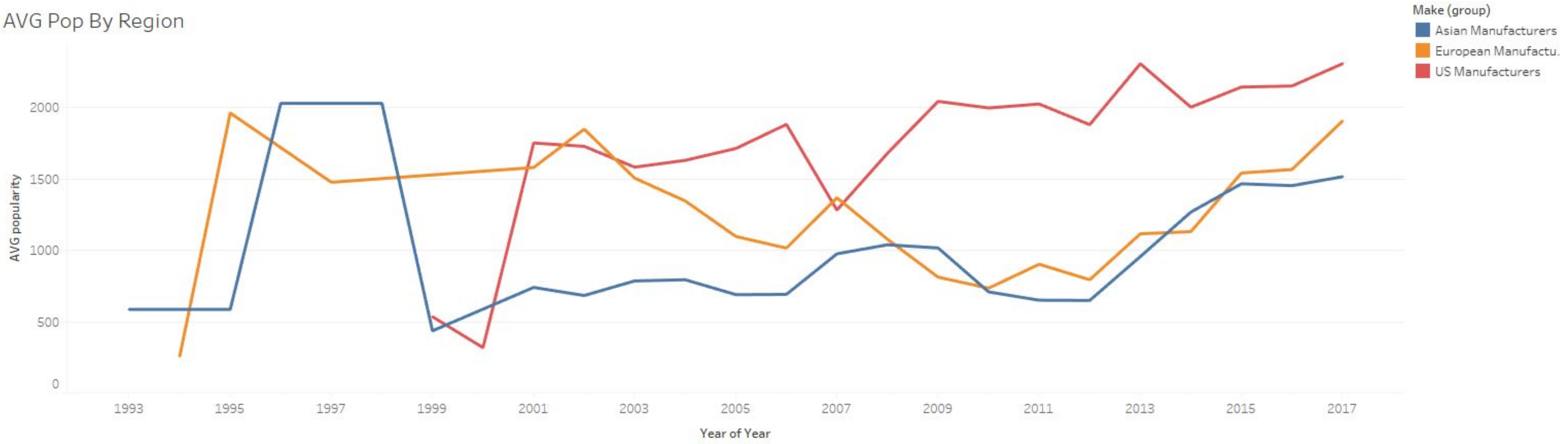

Numbers of cylinders

There are just a few more aspects of this data that I would like to explore and show before moving on to the next parts of this project. The chart above gives a relative sense of average price by category for every vehicle in the data. The information here is not surprising but it does show that categories one would expect to be more expensive such as ‘exotic’ and ‘factory tuner’ are indeed more expensive than models one would expect to be less expensive such as ‘standard’ and ‘hatchback’. We can also observe that most of the vehicles in the dataset have four-cylinder and six-cylinder engines, with eight-cylinder engines being the next most common. This information is not surprising, however it may be that any models that are created to predict MSRP may be better at predicting MSRP for the engine types that occur much more often than the others.

Finally, I would like to take a look at what is going on with the popularity data in this dataset. Since we can not actually be sure how exactly the popularity value was generated, it seems important to take some time to see what these values seem like they will be able to tell us. To be quite honest, the validity and usefulness of these values seems quite questionable, as some models have identical popularities for multiple years, which seems quite impossible if this was counting tweets for each model and each year, however, at the advice of my mentor I have not dropped the data, and will now present some of the information that the popularity data can provide.



This chart shows the most popular models in the dataset by the sum of their popularity across all years. It is important to note that this specific chart seems to be very misleading, as most of these models that have the highest popularity are also some of the most common models in the dataset, and it does not seem like the person that scraped this data made an effort to account for minor differences in models in the same year and their popularity value. On the next page, I've created a visualization that shows average popularity by year and region. Excluding the early data which has few samples so averages will tend to vary wildly there anyway, it seems that since 2011 the average popularity of all vehicles in all regions has seemed to increase over time. This makes intuitive sense since the popularity data was derived from Twitter, however it may be worth considering this popularity increase simply has to do with the number of people using Twitter as opposed to actual objective measure of popularity, since we do not know exactly how this particular value was measured. It will be interesting to see if the popularity values assist in predicting MSRP or simply add noise to any of the models that are created.



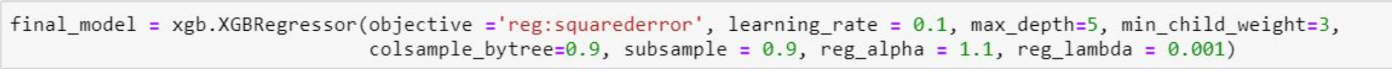
Model Building:

After performing all of my EDA, I moved to building a model to predict MSRP using the features that have been previously described in the dataset. I used XGBoost for this project on the advice of my mentor, knowing that it is the standard algorithm many companies use when they begin to build a model. Categorical features such as ‘make’ and ‘model’ were one hot encoded, and numerical features were standardized. I looked at the out of the box results of the model and the respective feature importances, for time constraints I will not speak much to the out of the box results as they were somewhat disappointing. After analyzing the results of the out of the box XGBoost model and discussing these results with my mentor, I performed a few janitorial tasks on the data for the sake of trying to improve model performance before tuning a final model for the project. These data modifications can be quickly described as follows:

1. The ‘make’ features were modified to include only the top 30 most common manufacturers, with the remaining manufacturers being labeled as ‘other’. Bugatti data was removed entirely as the three vehicles this manufacturer contributed to the dataset were clearly extreme outliers.
2. The ‘model’ features were similarly modified, keeping only the top 30 most common models and each other model being labeled as ‘other’.
3. Missing horsepower values were filled in with the mean horsepower value.

Model Tuning & Feature Importance Analysis:

With the data in a better state than it was before, I moved forward with tuning an XGBoost model to attempt to minimize RMSE. I used three-fold cross-validation while tuning each of the individual parameters of the model. The tuning process can be observed more thoroughly in my coding notebook. The best parameters can be seen in the code image with the final model label on the next page. This new model has a

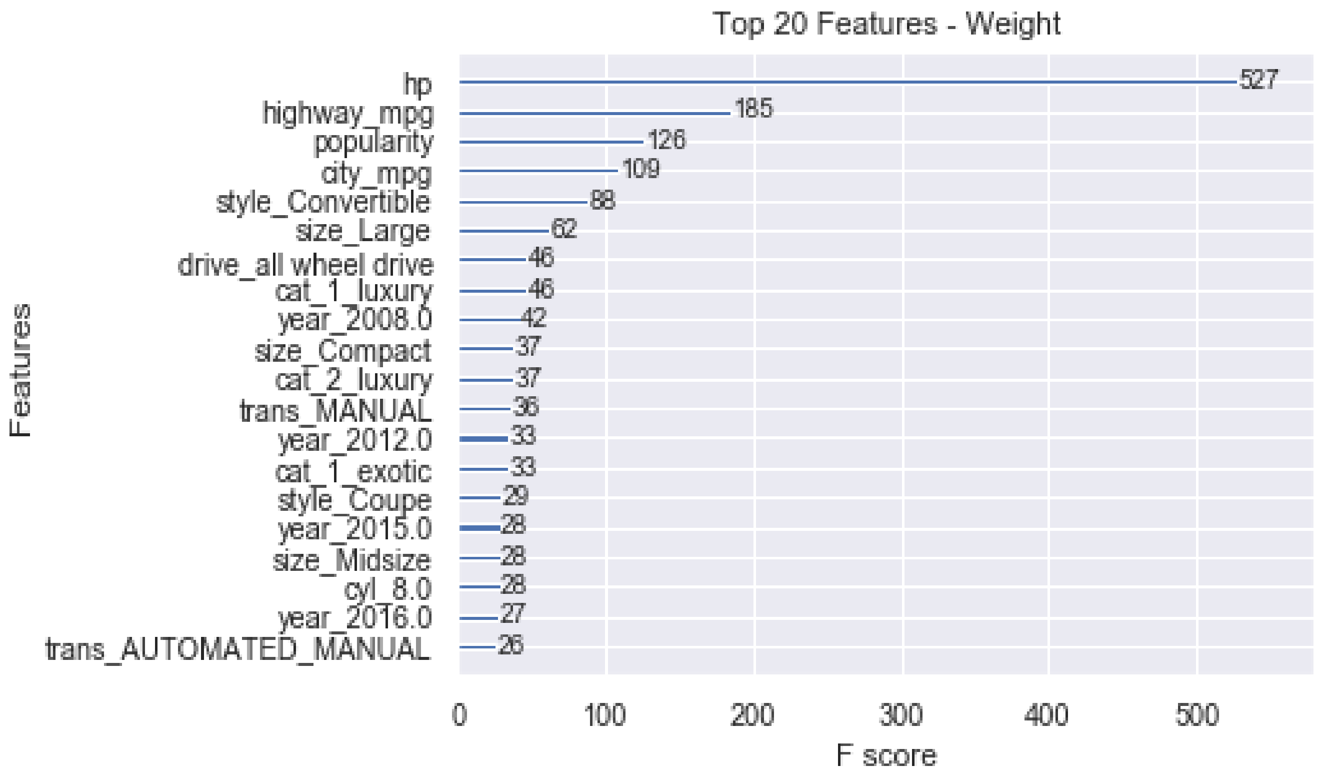


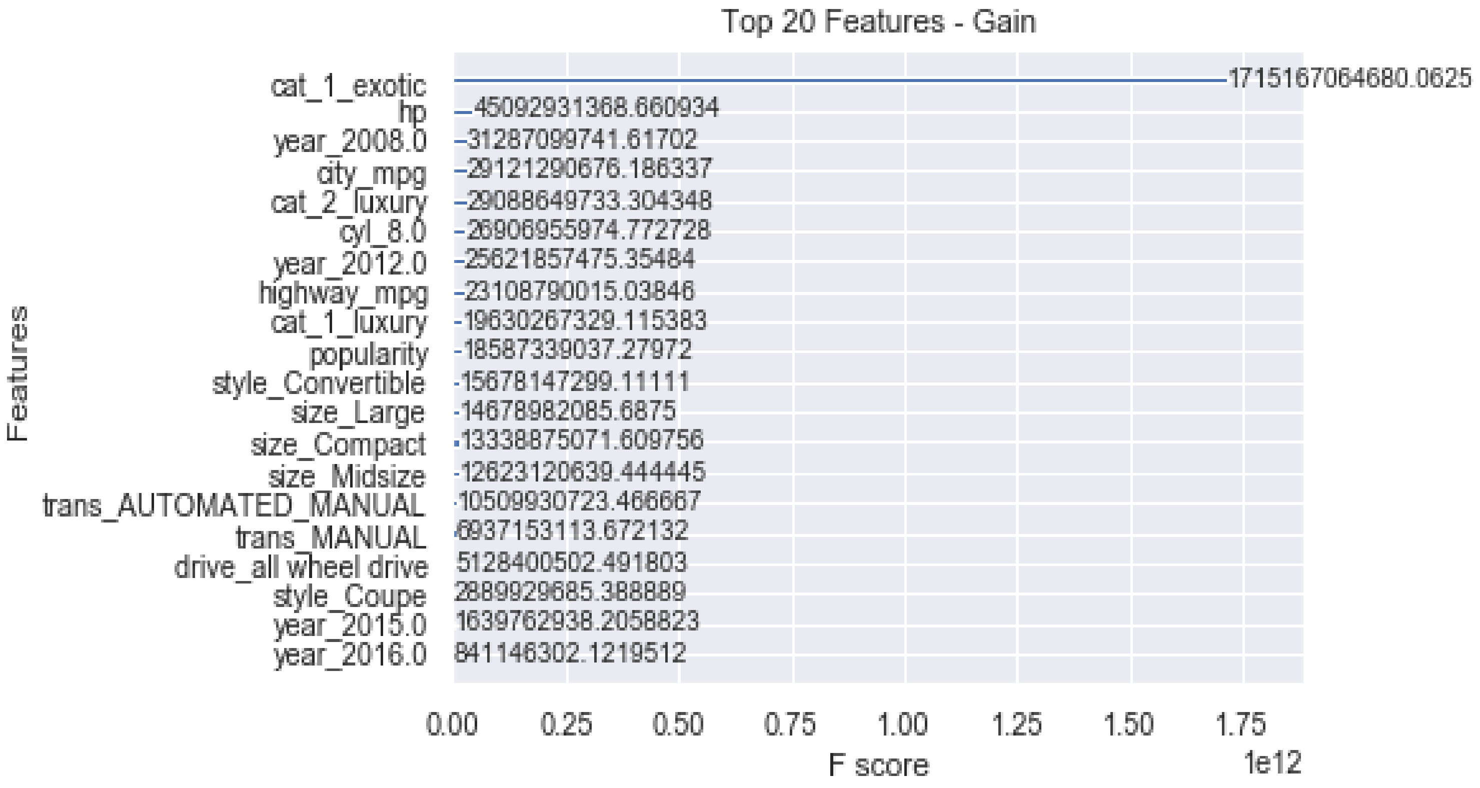
```
final_model = xgb.XGBRegressor(objective ='reg:squarederror', learning_rate = 0.1, max_depth=5, min_child_weight=3,
                               colsample_bytree=0.9, subsample = 0.9, reg_alpha = 1.1, reg_lambda = 0.001)
```

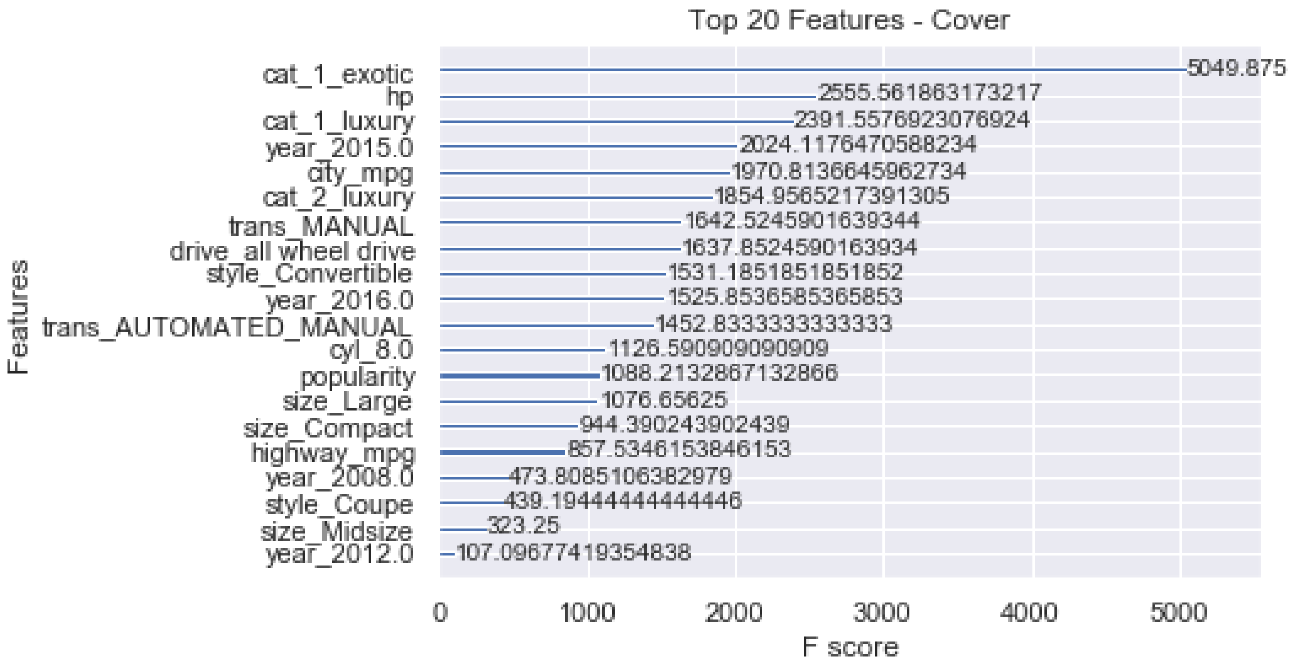
testing RMSE of approximately 12415.24, an improvement of the out of the box model performance of approximately 14507.32 by around 14.5%.

Now that the model was tuned it was time to start analyzing the feature importances of the model and pulling insights out of those feature importances. The following images are graphs that represent feature importance using the three measures of feature importance that XGBoost uses, which are: ‘weight’, ‘gain’, and ‘cover’. Each of these importances are described in the XGBoost documentation 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

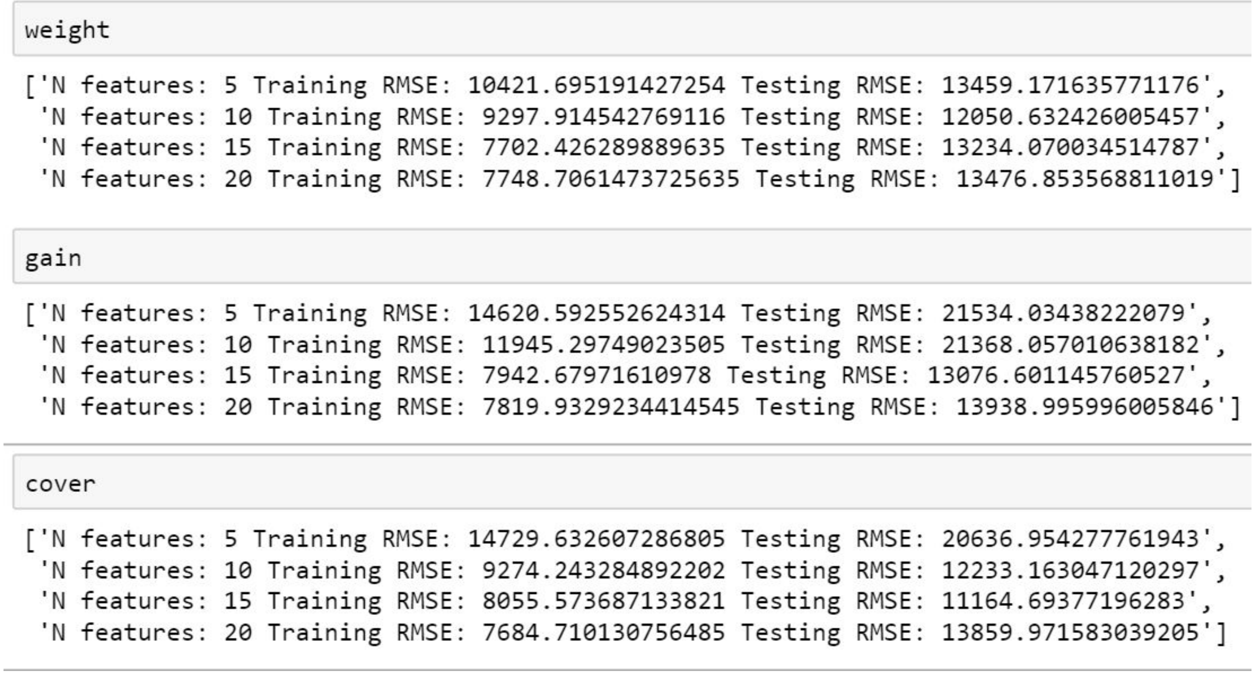

Top 20 Features - Weight
Features
F score
hp
highway_mpg
popularity
city_mpg
style_Convertible
size_Large
drive_all wheel drive
cat_1_luxury
year_2008.0
size_Compact
cat_2_luxury
trans_MANUAL
year_2012.0
cat_1_exotic
style_Coupe
year_2015.0
size_Midsize
cyl_8.0
year_2016.0
trans_AUTOMATED_MANUAL
527
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


Top 20 Features - Gain


Top 20 Features - Cover

Observing each of these charts we can see several common themes running through them. In each importance type, we see that whether a vehicle is labeled as ‘exotic’ or not is important. We also see that the ‘luxury’ category is important. The numerical features(horsepower, highway and city miles per gallon, and popularity) all seem to carry a higher degree of importance for each category, as they are all within the top 20 features for each importance type. The ‘manual’ and ‘automated manual’ transmission types both appear to be important as they also make multiple appearances in the top 20 of these importance types. Size seems to be an important feature for determining weight, as some size features appear in each importance list. Finally it seems pertinent to note the appearances of a few years in the dataset, specifically 2008, 2012, 2015, and 2016. The years 2015 and 2016 contain the largest numbers of records in the dataset, which is most likely the reason they are appearing here. 2008 and 2012 are less easy to explain, as it is not immediately apparent why they would be more important than other years in the dataset.

Now that we have observed the top 20 features of each importance type we can easily consider, I am going to look at the results of models that use only these top features to see how they perform against the tuned and cross-validated model to get an idea for how well these features predict MSRP against all of the features included in the data set. I have created a function that will help in reducing typing out different steps repetitively for this process. The code below is the result of using the top N features from each of the top 20 feature lists from the graphs above:



```
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']
```

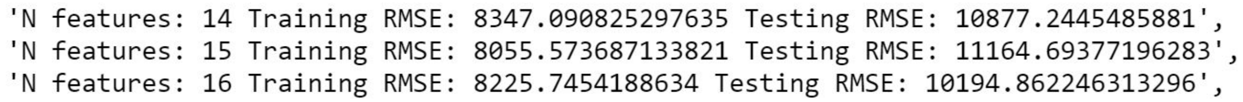
```
gain
```

```
['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
```

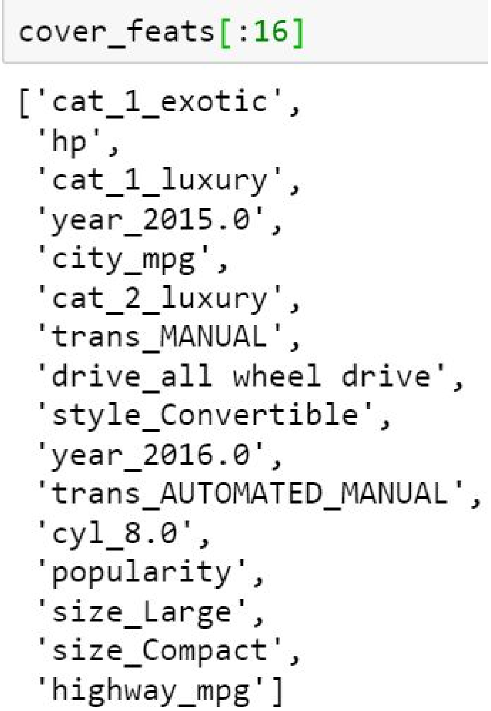
```
['N features: 5 Training RMSE: 14729.632607286805 Testing RMSE: 20636.954277761943',
 'N features: 10 Training RMSE: 9274.243284892202 Testing RMSE: 12233.163047120297',
 'N features: 15 Training RMSE: 8055.573687133821 Testing RMSE: 11164.69377196283',
 'N features: 20 Training RMSE: 7684.710130756485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

The first results that stand out to me are the results of: Weight-n=10, cover-n=10, and cover-n=15. Each of these feature sets generates an RMSE less than that of the model that includes all of the features. I then performed a more granular test around the numbers of features close to those numbers to see if the RMSE could be brought even lower by using a more specific set of features around those numbers. Upon checking the RMSE around these features, I discovered that the weight features generate a minimum RMSE at 10 features. The cover features, however, generated a minimum RMSE at 16 features, the lowest one that I observed during this project.



```
'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',
```

These were the three lowest RMSE values generated with this particular set of model specifications. We can see that the RMSE of the model with 16 features based on cover importance has an RMSE more than $2000 lower than the tuned best model, and more than $4000 lower than the out of the box RMSE. Now that we have discovered a model that seems to give us a minimized RMSE based on the procedures I have performed so far, we can look at these 16 features to see what they tell us about MSRP.



```
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']
```

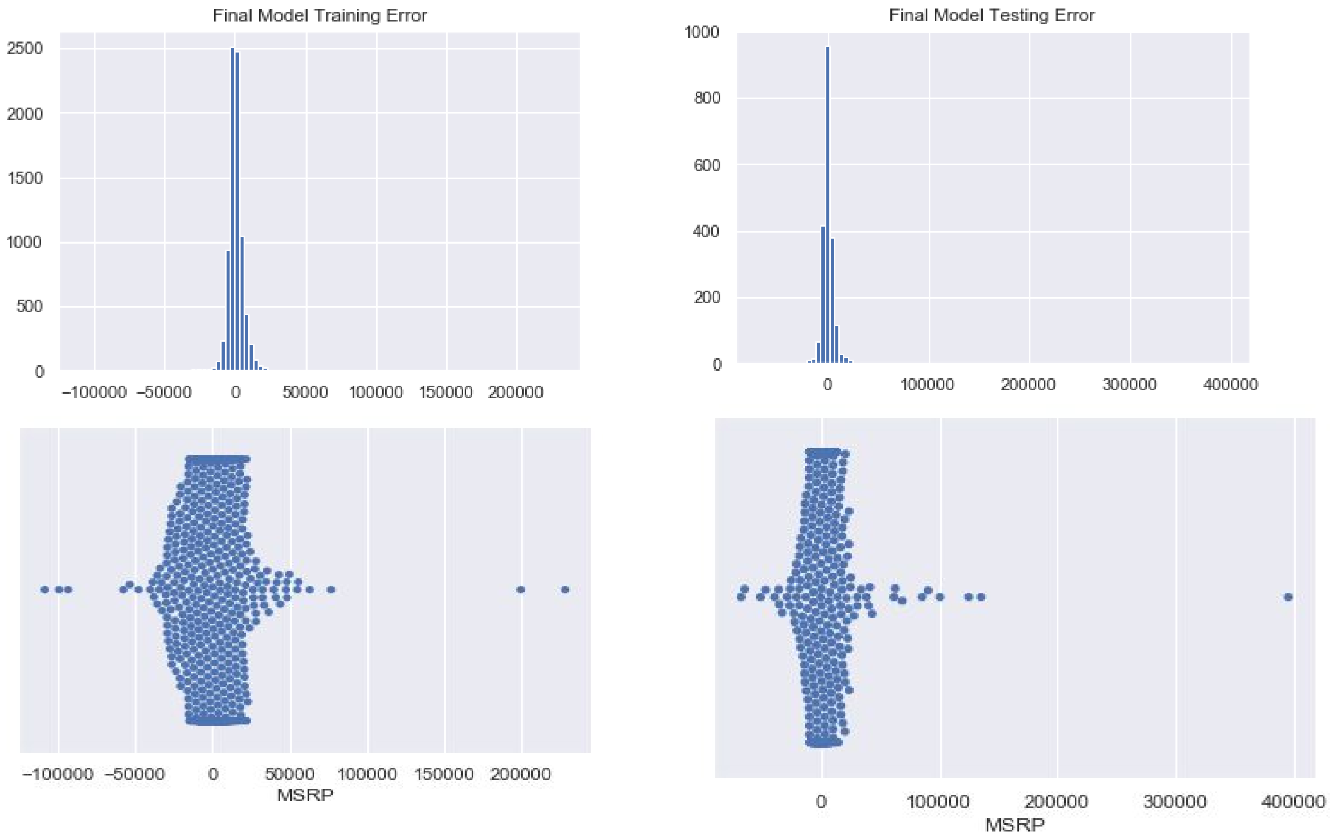
The image to the right is the list of features that generated the model with the lowest RMSE that I could find using the methods I have used for this project. There are a few points I would like to make about this particular list of features:

- All of the numeric features in the dataset are included in this list(hp, city_mpg, popularity, highway_mpg). Since we do not know exactly how popularity was measured, it may be pertinent to try to contact the creator of this dataset to get an idea for how exactly that feature was measured.
- The years 2015 and 2016 are present in this list. Since these were the most populated years in the dataset it is likely that these provided the model with the most information when it came to how a year affected price.
- Several categories that describe the most expensive cars in the dataset are in this feature list, specifically 'exotic' and both of the luxury categories are in this list.
  It seems that these features which could allow the model to differentiate between relatively expensive and inexpensive cars provides enough information to the model to be some of the more important features in the dataset.

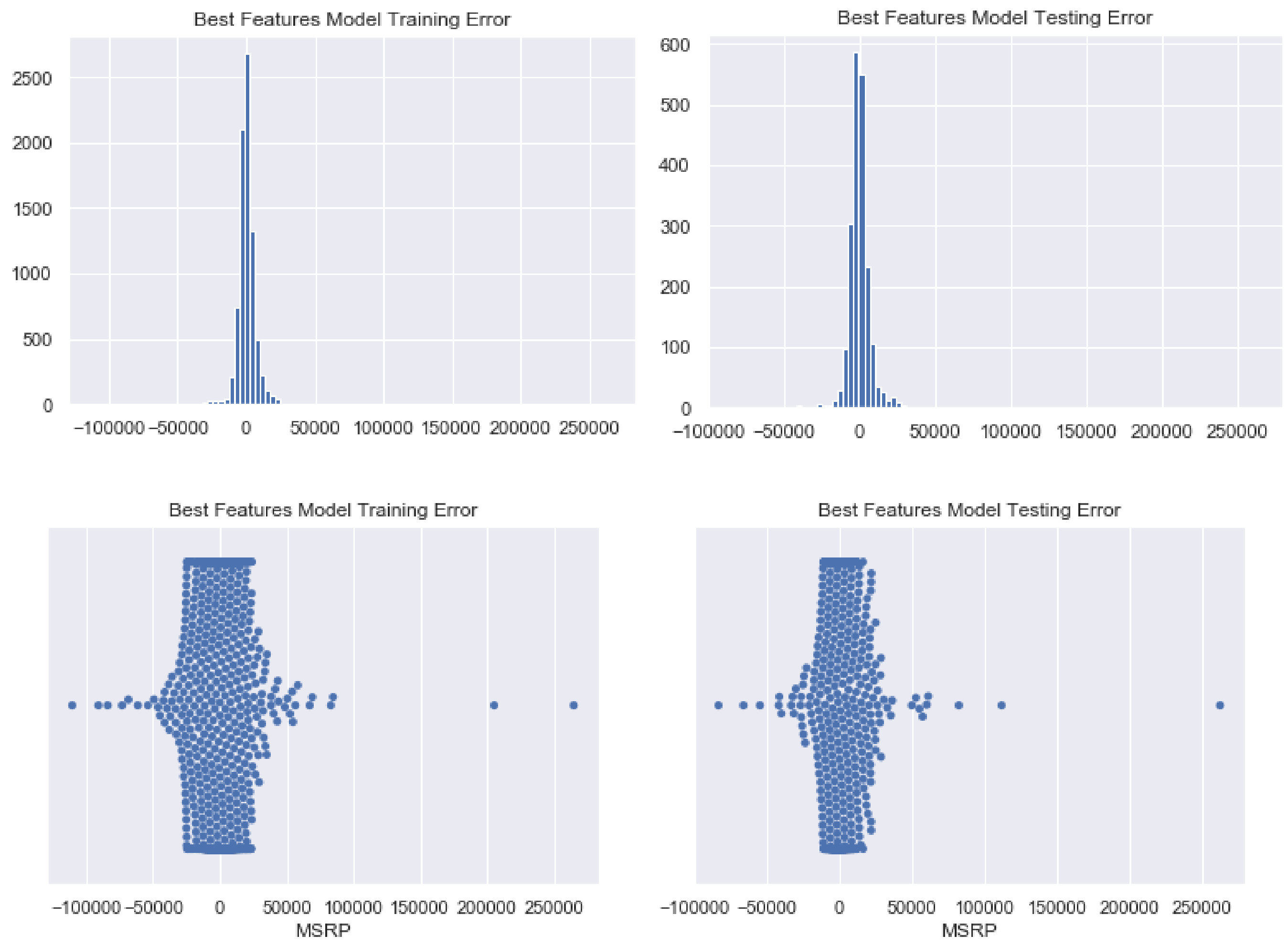
- Size appears to be an importance category for determining price as two of the three size features are in this list, this means all of the size information is contained within the model, since if a vehicle is not large or compact it is midsize.
- Manual and automated manual transmissions also seem to be important, and this seems to be another category group that could help the model differentiate between more expensive vehicles. The ‘convertible’ style and ‘all wheel drive’ category here also contains more information on expensive vehicles as well, as we observed earlier in this paper.

It seems that the most important features in this dataset are the features that assist a model in differentiating between the most expensive vehicles in the dataset, along with the numeric features. Now that the feature importance of the model has been analyzed, let’s take a deeper dive into the model results.

<u>Model Results:</u>

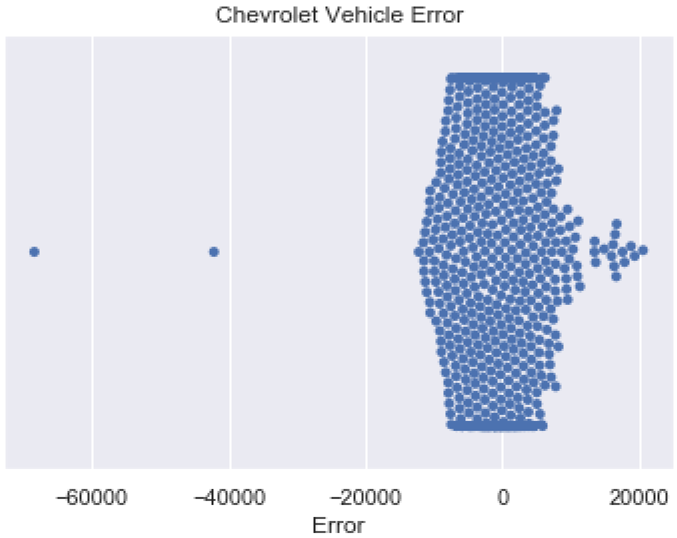


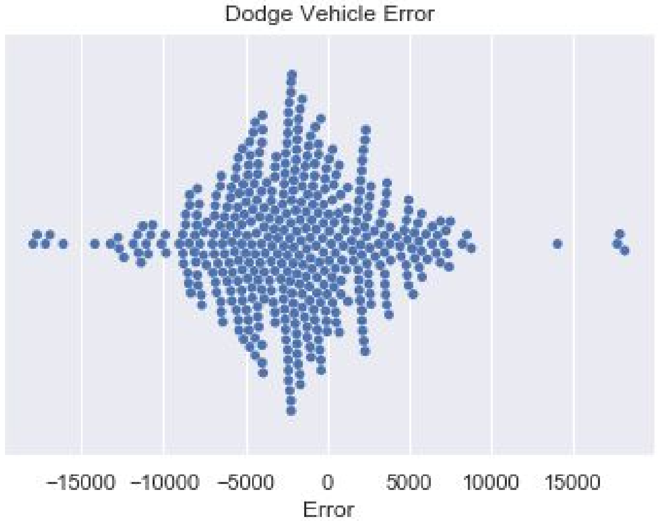
These graphs are plots of the error of the final model described before the feature importance analysis. It seems pertinent to compare these errors to the model that generates the smallest RMSE, so the following graphs show the errors of a model using only the 16 features discussed before.

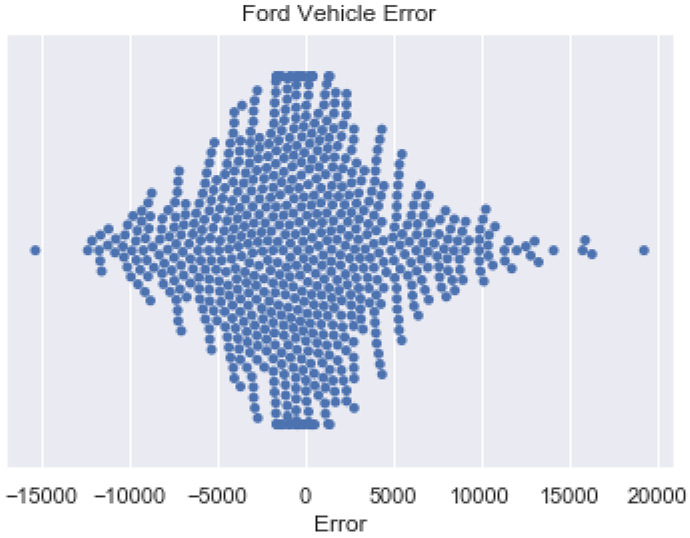


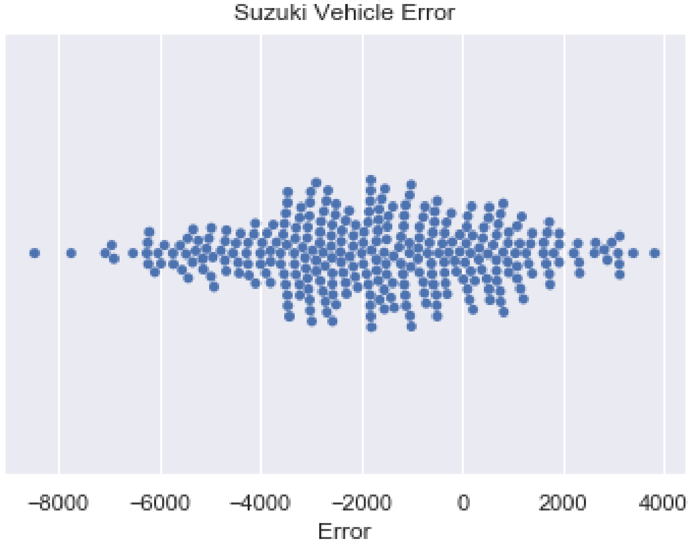
The first thing I see when looking at these errors is that the best features model has a much lower maximum testing error than the model containing all of the features. Beyond that it seems this particular model has a much larger error range than the model with all features. I do feel that looking at the errors of the entire dataset does not lead to much insight into what the model could be telling us, so I am going to spend some time reviewing what the model error on some of the most common makes and models look like, so see if anything is interesting in those areas. Before moving to that though, I would like to present the point that excluding some outliers, the error here does seem to be normally distributed, and the model does not seem to consistently overpredict or underpredict values.

Comparing Make Results:









Observing some of the results of some of the most prominent makes in the data, we can see that the model tends to underpredict some specific makes. I am not including all thirty of the top makes errors here, mainly for space and time reasons, and these four plots do a good job of showing what the model tends to do. As we can observe from the Ford and Chevrolet plots, the model will typically predict evenly around zero for a make, excluding some extreme outliers, but for some makes, such as Dodge and Suzuki, the center of the error mean is quite a bit below zero. I do not have the domain knowledge to make any empirical statements about these results, however it seems based on the features we have chosen to use in our model, Dodge and Suzuki may tend to underprice their vehicles relative to other manufacturers.

Final Thoughts and Moving Forward:

Were I to have more time right now to work on this particular project, I would like to spend more time looking at feature selection and perhaps engineering some new feature for the model, such as including region as a feature, or identifying a particular vehicle as something 'meant' to be cheap. The RMSE of the model seems to be rather large to me and I would have liked to get it even lower but currently I am at the limits of my knowledge when it comes to that. I think if I could perhaps build an ensemble model I may be able to get better results and lower the error of my model even more. I also wish that I could perhaps speak to the user that created the dataset to get a better idea of what the 'popularity' column actually was, and why so many of the cars from the earlier years in the dataset were overpriced.

When I set out to complete this project, I was hoping to find some clear answers for consumers as to what cars may be over or underpriced, but looking at these results it seems most manufacturers set their prices relatively normally, at least by the criteria the model looked at here. I think incorporating sales data into the project may be able to help in that regard, and perhaps one day if I ever feel the urge to return to that topic I will implement that into the model data as well if it is available.