BMW Pricing Challenge

Apoorva Jasti, Emily Wilkins, Matthew Zlotnik

Background

How do we assign a monetary value to a product? What is a “fair” rate? How do goods across different industries depreciate differently? As business analysts, these are questions we must consider. This can be a particularly challenging question when the good is a physical resource that loses value not only to gradual wear and tear, but a variety of other features as well. When a car model is standardized and straight from the assembly line, there is a sticker price assigned that follows a similar pattern. But after that point, it can be harder to nail down a car’s value. In fact, a new car loses value the moment is driven off the lot. Data collected by Carfax shows that cars can lose more than 10 percent of their value during the first month after you purchase it. In addition, a car can lose more than 20% of its value during the first year of ownership and then lose ~10% of its value every year for the next four years.[1]

While the actual utility of a car doesn’t decrease that quickly, its monetary value does in the eyes of the market. But beyond rough depreciation estimates, how do used car dealers price their cars? What is taken into consideration? We aim to answer that question in the following report using data from 4,843 BMW cars that were sold in a B2B auction in 2018.[2] While it is important to note that BMW is considered a luxury brand and its used vehicles may follow different patterns than an average used car, we think we still are able to derive some interesting insight into how used cars are valued.

Research Question

One of the everyday challenges in automotive business is estimating the value of a used car. Approximately 40 million used vehicles are sold each year, according to Edmunds.com, an online automotive review site.[3] The companies tracking those sales provide an invaluable resource with detailed information on what sells and for how much. Knowing the approximate used car pricing is important for sellers who need to unload their current vehicle, as it helps to ensure they properly price their car. This knowledge is equally important for buyers, allowing them to locate fairly-priced pre-owned vehicle options.

We believe that the sales price of a car is not only based on the value of the product itself based on its demographics, modifications, condition, and history, but is also heavily influenced by things like market trends, current availability, and politics. With this project, we hope to look more into this exciting topic and answer the research question: What are the main factors that drive the value of a used car?

EDA

Before we could begin running models on our data, we needed to glean a better understanding of the fundamental properties within the data. Generally, our data consisted of over 4,800 individual sales of BMW cars, and 18 attributes of the sale and car. These attributes included the date of sale, the model, mileage, engine power, registration date, color, fuel type, car type (including convertible, suv, sedan, etc.), and eight unnamed features which existed simply as binary true/false indicators of whether or not the car sold contained these features. To make a few of these features more model-friendly and to possibly add a bit of parsimony to our data, we subtracted the registration date field from the date of sale to get the age of the car when it was sold, and summed up the number of “true”s from the unnamed features to get a count of total added features for each car. To ensure that this was a legitimate process and that no two features were mostly sold together (which would mean that summing features would likely double count one possible package), we ran a correlation plot on the eight unnamed features. That plot is shown in Figure 1.

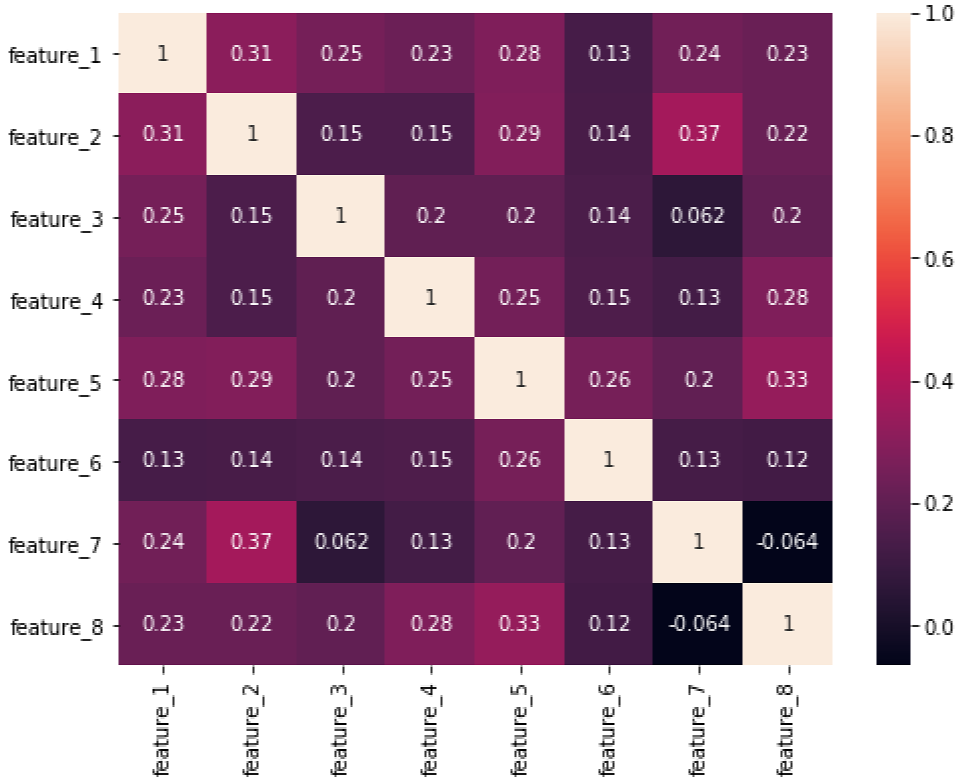

Figure 1: Correlation plot between all features

As you can see in the figure, no two features have a correlation greater than .37, so using a sum of the features as a proxy is a valid technique. Then, because we felt like model type was important, but knew most analytical models would not do well with a factor variable consisting of 75 different possible factors, we simplified all cars into the 14 types of models that BMW offers. These model types are 1-7, M, I, X, Z, Grand, and Active. Once we had engineered these features, we decided to check correlations between each of our continuous features to ensure we would not double count any level of the quality of the car. The correlation between vehicle age, engine power, mileage, and total features are shown in Figure 2 below. As you can see, engine power and age at sale, as well as mileage and age at sale were both pretty strongly correlated. As such, we elected to leave age at sale out of our models, believing that the mileage of the car would convey a more meaningful age of the car anyways.

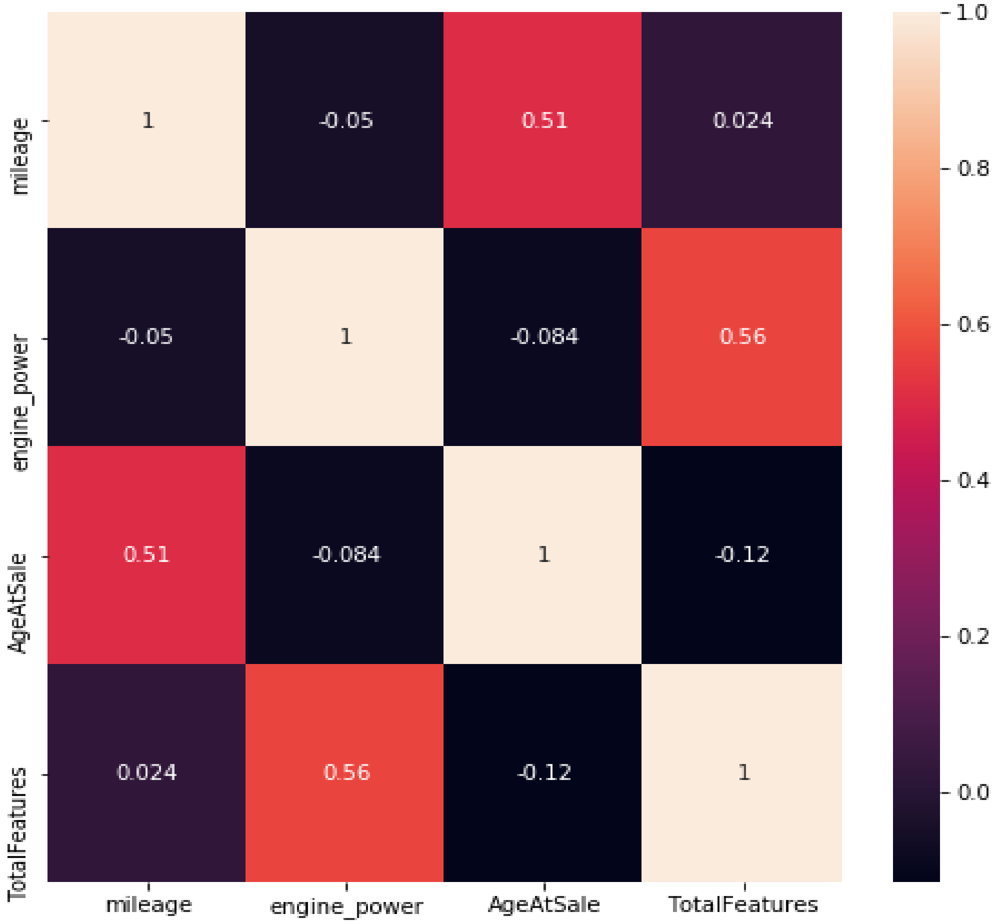

Figure 2: Correlation plot between mileage, engine power, age, and total features

Next, we decided to create pairwise plots of our predictors and the price of the car at sale to not only see if there were any further alterations that needed to be done to the data but also provide some backing to any initial hypothesis we could make. One of the more interesting findings of these plots was that hybrid cars tended to sell for considerably more on average than any other type of fuel. These results are shown in the figure below.

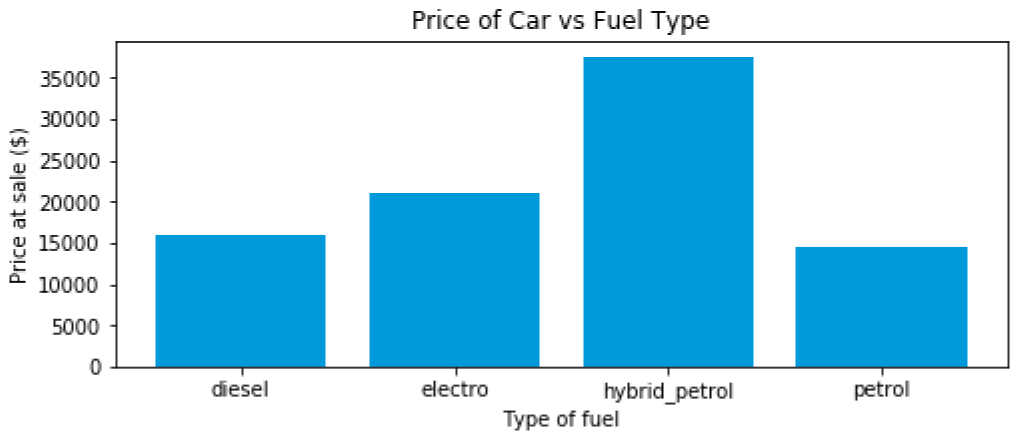

Figure 3: Price of Car vs. Fuel Type

Another insight we gleaned from this data was that the color of the car mattered little in the pricing of the car, which makes intuitive sense, but we were glad to see nonetheless. Possibly the most surprising insight, however, was that the total number of added features seemingly does not affect price much at all. In the figure below, you can see that there is no increase in price at sale relative to the number of added features present in the vehicle.

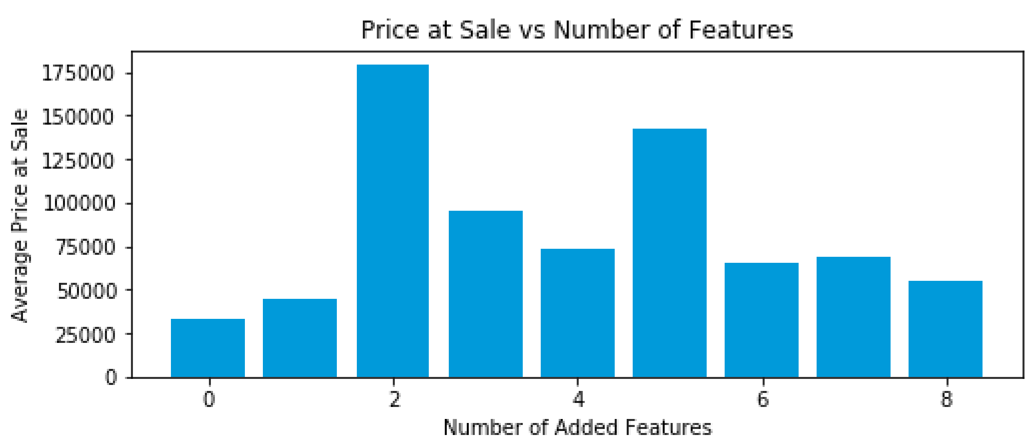

Figure 4: Total Features vs. Price

In fact, holding all else equal, it would appear as though having either two or five added features in a vehicle would increase the sale price the most. Finally, our other (mostly mundane) results from pairwise plotting were that price decreased with age and mileage of a car, while price increased with engine power.

Methodology

Clustering

To help gain more insight on the data we had, we elected to run a couple of basic clustering algorithms and see if they may add parsimony and possibly predictive power to our models. The first clustering technique we tried was DBSCAN, which looks for densely packed regions of the data, and separates them from other, less densely packed areas. As sparsity tends to increase with dimension, DBSCAN typically performs best on low-dimensional data. To fit DBSCAN on our data, we tried 100 combinations of epsilon (the maximum distance between two points to consider them densely packed) and minimum_points (the minimum number of points required to create a cluster. For some reason, even with 100 combinations of these two parameters, our DBSCAN algorithm was unable to differentiate any clusters, clustering the entire dataset as “noise” for all 100 iterations. Assuming maybe the wide range of mileages could have created too much distance between points, we decided to standardize all three of our remaining numerical features. This allowed DBSCAN to find clusters, but they tended to be wildly imbalanced, with noise still constituting between 50-90% of all data.

Deciding that DBSCAN likely would not add much value to our later predictive modeling, we moved on to k-means. For k-means, we simply had to test for the correct number of clusters to create. As you can see in the SCREE plot in Figure 5, the two main elbows of the plot are at k = 4 and k = 6. Because the elbow at k = 4 appears to be significantly sharper, and the score does not increase considerably from there, we decided that k= 4 would be our most effective k-means clustering parameter. With this new cluster assignment value attached to the data, we moved on to supervised learning methods.

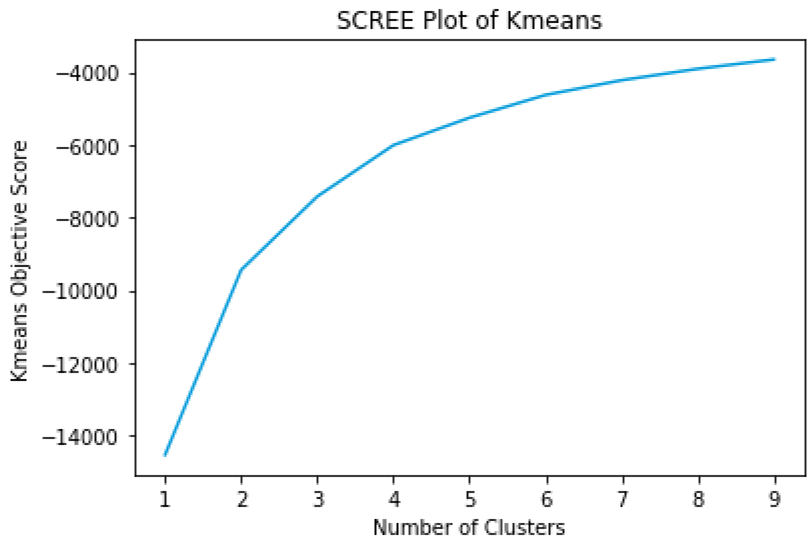

Figure 5: Scree Plot

Linear Regression

We started with one of the simplest supervised methods, a multiple linear regression, to establish a baseline . Our initial regression using all the variables except the maker key since all the vehicles are BMWs, the model key since we have elected to use the feature we engineered called model type instead, and the age of the vehicle since it correlates highly with mileage and we believe mileage is the more informative feature. However, we run a regression including age and excluding mileage to confirm this. Before modeling, we also standardized all the discrete numeric columns in the data set, namely mileage, engine power, age, and total features, and changed the features fuel, paint color, car type, and model type to factors.

The results of this initial regression model are shown in Figure A in the Appendix. From this model we find that mileage, engine power, registration date, fuel type, some car types, some model types, and all the features except Feature 5 seem to be significant predictors of price. Color and total features do not seem to matter, or rather total features correlates too highly with the features themselves, so it is from the regression. This model has an R-squared of 0.6848 and adjusted R-squared of 0.6819. To confirm that mileage is more informative than the age of the vehicle when predicting price, we ran nearly the same regression as shown in Figure A, but this time excluded mileage and included age at sale. The results of this regression model are shown in Figure B in the Appendix. As suspected, we do get more information from including mileage than from including age as shown by the lower R-squared value of 0.652. In addition, age does not even show as being significant. Therefore, going forward, we will not include age as a feature.

With the results from these regressions in mind, we run another linear regression that models on the same subset of the data as in the initial regression, but also excludes the variables found to be insignificant, namely paint color and total features. The results from this regression can be found in Figure C in the Appendix. We see that when we re-run the initial regression excluding the insignificant variables we lose very little predictive power. R-squared decreases by 0.007 and the adjusted R-squared only decreases by 0.001. This model achieves nearly the same accuracy while simplifying the model by eliminating 10 features.

Principal Components Regression

After establishing this baseline accuracy, we try to see if we can improve upon these results through a principal components regression. Principal component analysis (PCA) allows us to reduce the dimensionality of our data set while preserving as much of the variance of the original data as possible. To prevent the algorithm from being skewed towards predictors that are large in absolute scale but potentially not as relevant as others, it is important to standardize the data before running PCA. Since we did this before modeling, we can skip this step. PCA also only works on numerical data, so we can only include numeric predictors in this analysis. Using the same data set as in the final regression model, we only need to exclude model type before using the principal components regression function. The results from the principal components analysis are shown in Figure 6 below. Twenty two principal components are considered, one for each feature.

```
Data:   X dimension: 4843 22
        Y dimension: 4843 1
Fit method: svdpc
Number of components considered: 22

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
CV            9221     7091     6089     6010     5876     5810     5788     5775     5776     5776      5786      5769      5692      5692      5694      5682
adjCV         9221     7091     6089     6010     5875     5809     5787     5774     5775     5774      5783      5767      5690      5690      5691      5679
       16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
CV         5679      5638      5538      5520      5325      5323      5321
adjCV      5677      5635      5536      5518      5322      5319      5318

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps
X        13.97    22.46    29.74    36.25    42.44    48.20    53.19    57.92    62.50     66.99     71.30      75.3     79.18     82.84     86.27     89.22
price    40.89    56.54    57.68    59.58    60.51    60.83    61.05    61.06    61.29     61.29     61.43      62.4     62.44     62.48     62.64     62.68
       17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
X         91.96     94.55     96.62     98.36     99.95    100.00
price     63.25     64.59     64.81     67.35     67.38     67.43
```

Figure 6: Principal Components Analysis

We selected the number of principal components to retain based on the graph in Figure 7. We find that R-squared levels off around 4 principal components, so we opt to regress on only the first four principal components.

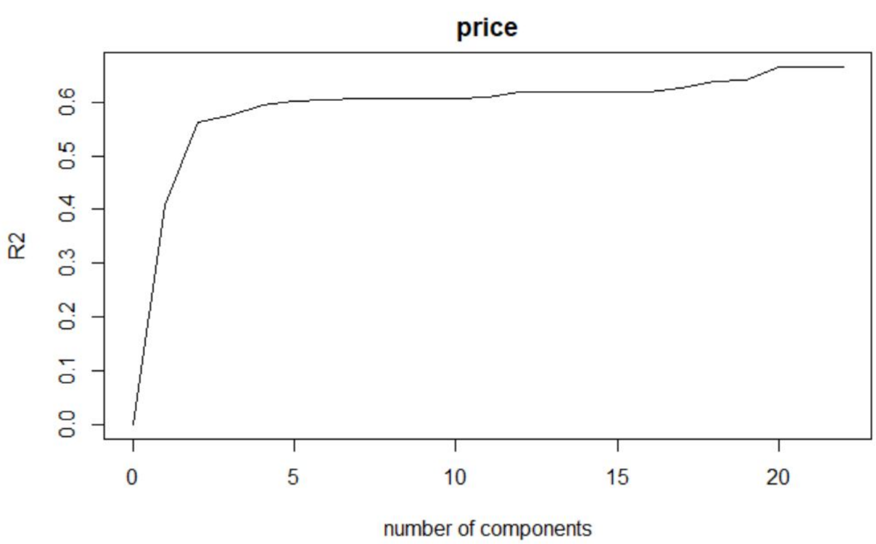

Figure 7: R-squared vs. Number of Principal Components

To be able to compare the results of our principal component regression against a multiple linear regression, we split the subset of the data used in finding the principal components into a train and test set. Using that train and test set, we calculate the out-of-sample mean squared error (MSE) for both models. We find the test MSE for the principal component regression is 26,920,358 while the test MSE for the multiple linear regression is 21,080,932. This shows that the principal component regression's MSE is greater than that of the linear regression. This indicates that performing principal component analysis on this data does not improve the predictive power of our model. With these results in mind, we move onto other model selections to try to improve our predictions.

Decision Trees, Random Forest, and Boosting

We then decided to look into using trees to predict the price of the model. We decided to use decision trees, random forest and boosting algorithms.

Feature engineering done for the dataset used in the following algorithms are:

- Standardized the discrete numeric columns - Mileage, Engine power, Total features, Age at sale
- One hot encoded the categorical columns - Fuel, Car type, Car color

We built decision trees using all the features, MSE as criterion and different variations of max_depth. We got the best adjusted R-squared as 0.59 which was slightly lower than regression. We then decided to use a random forest model. Random forest models are better than decision trees as they consist of multiple single trees each based on a random sample of the training data leading to reducing overfitting and error due to bias. We built 50 estimators using mse as a criterion and different variations of max depth. Using this model, we saw an adjusted R-squared of 0.8 which was an improvement from the decision tree model. We also built a variable importance plot to understand which features were most important. From Figure 8**,** we can see that engine power, age of the car and mileage are the most important features to predict price.

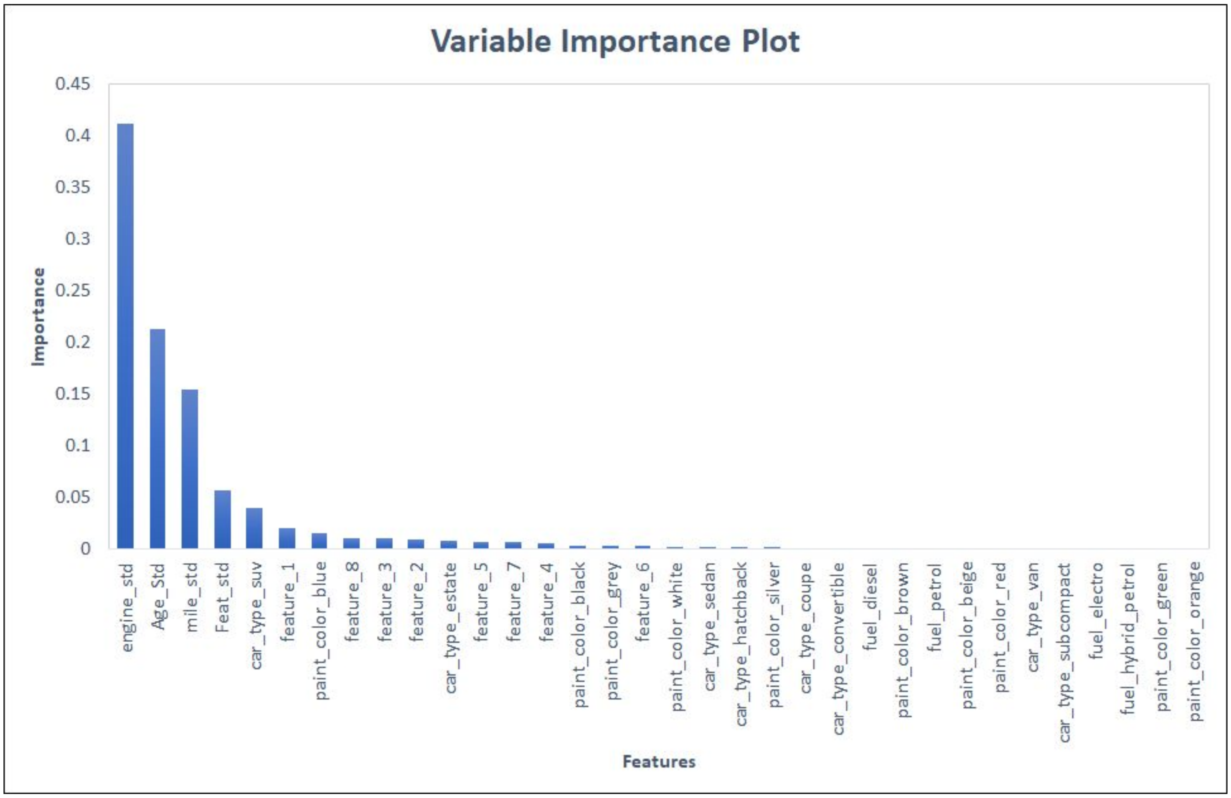

Figure 8: Feature Importance

Post this we decided to try a boosting algorithm. Gradient boosting Algorithm reduces error mainly by reducing bias (and also to some extent variance, by aggregating the output from many models). We used a gradient boosting regressor with parameters set as min leaf samples= 15, learning_rate=0.1, max_depth=4, criterion = MSE and estimators=100 and got an adjusted R-squared as 0.82.

Conclusion

We saw that the engine power, mileage, and age are the most important factors affecting the sale price of a car similar to our assumptions. We also saw the feature 1, 2, 3 and 8 have a higher impact on the price of the car compared to other features. It would be helpful having those details so as to explain the exact details. Among the different car types, hybrid cars tend to sell at a higher cost compared to other models.

Next Steps

We would like to look at how the estimated value of a car changes over time (e.g. the price of a convertible should be higher in summer than in winter) and the effect of geographical location on sale value.

References

1. Krome, Charles. “Car Depreciation: How Much Value Will a New Car Lose?” *CARFAX*, 5 Feb. 2019, www.carfax.com/blog/car-depreciation.
2. Kirch, Daniel. “BMW Pricing Challenge.” *Kaggle*, 7 Mar. 2019, www.kaggle.com/danielkyrka/bmw-pricing-challenge.
3. D'Allegro, Joe. “Just What Factors Into The Value Of Your Used Car?” *Investopedia*, Investopedia, 25 Feb. 2020, www.investopedia.com/articles/investing/090314/just-what-factors-value-your-used-car.asp.

Appendix

Figure A: Initial Regression

```
Call:
lm(formula = price ~ ., data = pricing_data2)

Residuals:
   Min     1Q Median     3Q    Max 
-25875  -2145   -120   1726 157946 

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -8.058e+04  2.084e+04  -3.867 0.000112 ***
mileage            -2.090e+03  9.329e+01 -22.405  < 2e-16 ***
engine_power        3.893e+03  1.133e+02  34.366  < 2e-16 ***
registration_date   2.669e+00  1.102e-01  24.221  < 2e-16 ***
fuelelectro        -1.308e+04  5.008e+03  -2.612 0.009038 ** 
fuelhybrid_petrol   6.854e+03  3.021e+03   2.269 0.023343 *  
fuelpetrol         -1.308e+03  4.264e+02  -3.069 0.002163 ** 
paint_colorblack    8.018e+02  8.309e+02   0.965 0.334611    
paint_colorblue     3.362e+02  8.437e+02   0.398 0.690330    
paint_colorbrown    7.999e+02  8.684e+02   0.921 0.357041    
paint_colorgreen    7.699e+02  1.489e+03   0.517 0.605221    
paint_colorgrey     7.928e+02  8.345e+02   0.950 0.342097    
paint_colororange  -1.487e+03  2.324e+03  -0.640 0.522245    
paint_colorred      1.366e+03  1.097e+03   1.246 0.212923    
paint_colorsilver   2.274e+02  8.698e+02   0.261 0.793747    
paint_colorwhite    9.117e+02  8.528e+02   1.069 0.285136    
car_typecoupe      -1.092e+03  9.930e+02  -1.100 0.271574    
car_typeestate     -3.592e+03  9.447e+02  -3.802 0.000145 ***
car_typehatchback  -2.742e+03  9.659e+02  -2.839 0.004546 ** 
car_typesedan      -1.530e+03  9.398e+02  -1.628 0.103576    
car_typesubcompact -1.421e+03  1.080e+03  -1.316 0.188261    
car_typesuv         9.455e+02  2.375e+03   0.398 0.690634    
car_typevan        -6.837e+03  1.415e+03  -4.831 1.40e-06 ***
feature_1TRUE       1.136e+03  1.791e+02   6.345 2.43e-10 ***
feature_2TRUE       5.276e+02  2.263e+02   2.331 0.019777 *  
feature_3TRUE       9.164e+02  2.030e+02   4.515 6.49e-06 ***
feature_4TRUE       1.429e+03  2.458e+02   5.813 6.54e-09 ***
feature_5TRUE      -2.720e+02  1.780e+02  -1.528 0.126525    
feature_6TRUE       8.457e+02  1.895e+02   4.462 8.31e-06 ***
feature_7TRUE       8.789e+02  3.504e+02   2.508 0.012161 *  
feature_8TRUE       1.553e+03  1.854e+02   8.374  < 2e-16 ***
sold_at             2.926e+00  1.172e+00   2.496 0.012576 *  
TotalFeatures              NA         NA      NA       NA    
ModelType2          7.470e+02  1.411e+03   0.529 0.596495    
ModelType3          4.468e+02  5.288e+02   0.845 0.398120    
ModelType4          4.000e+03  1.030e+03   3.882 0.000105 ***
ModelType5          1.434e+03  5.621e+02   2.551 0.010786 *  
ModelType6          7.756e+01  1.704e+03   0.046 0.963706    
ModelType7          5.171e+03  9.201e+02   5.620 2.02e-08 ***
ModelTypeActive     4.081e+03  1.680e+03   2.429 0.015167 *  
ModelTypeGran       2.974e+03  4.213e+02   7.060 1.91e-12 ***
ModelTypeI          1.918e+04  4.012e+03   4.781 1.80e-06 ***
ModelTypeM          3.498e+03  1.228e+03   2.848 0.004414 ** 
ModelTypeOTHER     -1.314e+04  6.044e+03  -2.175 0.029708 *  
ModelTypeX          4.100e+02  2.242e+03   0.183 0.854927    
ModelTypeZ         -2.755e+02  2.374e+03  -0.116 0.907592    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5200 on 4798 degrees of freedom
Multiple R-squared:  0.6848,	Adjusted R-squared:  0.6819 
F-statistic: 236.9 on 44 and 4798 DF,  p-value: < 2.2e-16
```

Figure B: Regression including age at sale

```
Call:
lm(formula = price ~ ., data = pricing_data2.1)

Residuals:
   Min     1Q Median     3Q    Max 
-25114  -2312   -169   1870 158885 

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -314031.11  150658.20  -2.084  0.03718 *  
engine_power          3923.21     119.02  32.963  < 2e-16 ***
registration_date     -101.76      75.35  -1.350  0.17693    
fuelelectro         -11267.60    5261.61  -2.141  0.03229 *  
fuelhybrid_petrol     7145.83    3175.15   2.251  0.02446 *  
fuelpetrol              93.67     443.20   0.211  0.83263    
paint_colorblack       380.82     872.89   0.436  0.66266    
paint_colorblue        104.67     886.55   0.118  0.90603    
paint_colorbrown       601.46     912.43   0.659  0.50981    
paint_colorgreen      1085.57    1564.88   0.694  0.48790    
paint_colorgrey        520.66     876.75   0.594  0.55264    
paint_colororange    -1858.94    2441.92  -0.761  0.44654    
paint_colorred        1161.41    1152.30   1.008  0.31355    
paint_colorsilver       33.79     914.03   0.037  0.97051    
paint_colorwhite       421.82     895.87   0.471  0.63777    
car_typecoupe        -1419.95    1043.60  -1.361  0.17370    
car_typeestate       -4379.70     992.07  -4.415 1.03e-05 ***
car_typehatchback    -3271.49    1014.63  -3.224  0.00127 ** 
car_typesedan        -1905.12     987.45  -1.929  0.05375 .  
car_typesubcompact   -2142.45    1134.30  -1.889  0.05898 .  
car_typesuv            409.31    2496.08   0.164  0.86975    
car_typevan          -6276.64    1487.19  -4.220 2.48e-05 ***
feature_1TRUE         1282.63     188.13   6.818 1.04e-11 ***
feature_2TRUE          201.50     237.33   0.849  0.39591    
feature_3TRUE         1032.82     213.22   4.844 1.31e-06 ***
feature_4TRUE         1557.22     258.21   6.031 1.75e-09 ***
feature_5TRUE         -477.60     186.76  -2.557  0.01058 *  
feature_6TRUE          836.56     199.25   4.199 2.73e-05 ***
feature_7TRUE          257.85     367.04   0.702  0.48240    
feature_8TRUE         1408.09     194.72   7.231 5.54e-13 ***
sold_at                108.97      75.34   1.446  0.14811    
AgeAtSale           -98017.07   69869.59  -1.403  0.16072    
TotalFeatures              NA         NA      NA       NA    
ModelType2            1065.31    1482.61   0.719  0.47246    
ModelType3            -318.94     554.47  -0.575  0.56517    
ModelType4            4248.39    1083.42   3.921 8.93e-05 ***
ModelType5             502.58     589.09   0.853  0.39363    
ModelType6            1016.15    1790.48   0.568  0.57038    
ModelType7            4783.84     967.28   4.946 7.85e-07 ***
ModelTypeActive       2633.35    1764.25   1.493  0.13560    
ModelTypeGran         2214.55     441.32   5.018 5.41e-07 ***
ModelTypeI           19452.55    4216.18   4.614 4.06e-06 ***
ModelTypeM            2761.19    1289.96   2.141  0.03236 *  
ModelTypeOTHER      -12208.50    6351.36  -1.922  0.05464 .  
ModelTypeX             106.94    2356.42   0.045  0.96381    
ModelTypeZ            1035.41    2493.39   0.415  0.67797    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5464 on 4798 degrees of freedom
Multiple R-squared:  0.652,     Adjusted R-squared:  0.6488 
F-statistic: 204.3 on 44 and 4798 DF,  p-value: < 2.2e-16
```

Figure C: Linear regression excluding insignificant features

```
Call:
lm(formula = price ~ ., data = pricing_data4)

Residuals:
   Min     1Q Median     3Q    Max 
-25791  -2124   -123   1734 157577 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -7.900e+04  2.078e+04  -3.801 0.000146 ***
mileage            -2.079e+03  9.309e+01 -22.334  < 2e-16 ***
engine_power        3.894e+03  1.130e+02  34.468  < 2e-16 ***
registration_date   2.699e+00  1.081e-01  24.959  < 2e-16 ***
fuelelectro        -1.311e+04  5.006e+03  -2.619 0.008843 ** 
fuelhybrid_petrol   6.781e+03  3.021e+03   2.244 0.024851 *  
fuelpetrol         -1.256e+03  4.258e+02  -2.949 0.003207 ** 
car_typecoupe      -1.125e+03  9.895e+02  -1.136 0.255826    
car_typeestate     -3.679e+03  9.404e+02  -3.912 9.29e-05 ***
car_typehatchback  -2.796e+03  9.620e+02  -2.906 0.003677 ** 
car_typesedan      -1.618e+03  9.358e+02  -1.729 0.083907 .  
car_typesubcompact -1.537e+03  1.076e+03  -1.428 0.153258    
car_typesuv         8.864e+02  2.375e+03   0.373 0.708968    
car_typevan        -6.848e+03  1.413e+03  -4.847 1.29e-06 ***
feature_1TRUE       1.129e+03  1.782e+02   6.337 2.56e-10 ***
feature_2TRUE       5.062e+02  2.257e+02   2.243 0.024955 *  
feature_3TRUE       9.243e+02  2.028e+02   4.557 5.32e-06 ***
feature_4TRUE       1.405e+03  2.452e+02   5.732 1.05e-08 ***
feature_5TRUE      -2.520e+02  1.773e+02  -1.421 0.155286    
feature_6TRUE       8.621e+02  1.888e+02   4.566 5.09e-06 ***
feature_7TRUE       8.906e+02  3.491e+02   2.551 0.010772 *  
feature_8TRUE       1.548e+03  1.853e+02   8.354  < 2e-16 ***
sold_at             2.853e+00  1.169e+00   2.440 0.014738 *  
ModelType2          6.161e+02  1.406e+03   0.438 0.661198    
ModelType3          4.824e+02  5.276e+02   0.914 0.360597    
ModelType4          3.962e+03  1.030e+03   3.845 0.000122 ***
ModelType5          1.467e+03  5.607e+02   2.616 0.008928 ** 
ModelType6         -7.194e+01  1.702e+03  -0.042 0.966290    
ModelType7          5.251e+03  9.189e+02   5.714 1.17e-08 ***
ModelTypeActive     4.016e+03  1.677e+03   2.395 0.016677 *  
ModelTypeGran       2.942e+03  4.186e+02   7.027 2.40e-12 ***
ModelTypeI          1.937e+04  4.011e+03   4.829 1.41e-06 ***
ModelTypeM          3.317e+03  1.225e+03   2.708 0.006791 ** 
ModelTypeOTHER     -1.294e+04  6.036e+03  -2.144 0.032051 *  
ModelTypeX          4.320e+02  2.242e+03   0.193 0.847229    
ModelTypeZ         -6.103e+02  2.348e+03  -0.260 0.794926    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5201 on 4807 degrees of freedom
Multiple R-squared:  0.6841,	Adjusted R-squared:  0.6818 
F-statistic: 297.4 on 35 and 4807 DF,  p-value: < 2.2e-16
```