# Predictive Analysis of Used Vehicles

**Team Members**: Gabrielle Beinars, Daniel Wiltse, Dan Zylkowski

## Abstract/Executive Summary

Used cars are the biggest segment of car sales today, as the market is twice as big as the new car sales market and has grown even bigger due to decreased manufacturing of new vehicles during the global pandemic. With increased competition for used cars, it is essential to ensure the consumer is getting a fair value for the automobile being purchased. Understanding the fair price for a used vehicle is beneficial for the dealership as well as the consumer.

Seven datasets containing information on various vehicles, including Audi, BMW, Mercedes, Volkswagen, Toyota, Hyundai and Ford, were explored to identify patterns that provide insight for the dealer and consumer. Variables included in the datasets and used in the analysis are model, year, transmission, mileage, fuel type, road tax, miles per gallon (mpg), and engine size, with price as the target variable. Once the datasets were combined, the vehicle make/manufacturer was added, and various exploratory data analysis functions were created and used to preprocess the data.

Using several different predictive modeling techniques, including linear regression, random forest, and Catboost Regessor, a baseline model was built that predicted used car prices with $R^2$ of 0.9548. $R^2$ , also known as the coefficient of determination, is a metric used in regression to measure goodness of fit, and how much variation in price can be explained by the independent variables.[8] In addition, RMSLE, root mean squared log error, can also be used as an evaluation metric where the lower the value, the better, and outliers are drastically scaled down.[9] We further enhanced our model using feature engineering as well as web scraping reliability rating data in efforts to improve our models' performance. Using the additional features, we were able to increase the $R^2$ value to 0.958, and using ensemble technique were able increase the $R^2$ nearly 1% from baseline, to 0.964. While satisfied with overall results of analysis, limitations of dataset and modeling techniques are also discussed.

Keywords: *Predictive Analytics, Machine Learning, Regression Model, PyCaret*

## Introduction

### Background of Problem

Used cars are an essential part of the automobile sales ecosystem. Consumers purchase used cars to save on a new car's premium price, and car dealerships use their off-lease and trade-in inventory to satisfy the demand. Adding to the dynamic nature of the marketplace, the Covid-19 pandemic has created a boom in the used car market. The boom is the result of both a drop in new vehicle production and because people have purchased cars to try to avoid the risks associated with mass transit [1]. The result is that car dealers have to purchase additional vehicles to meet the increased demand.

### Problem Statement

Understanding the fair market price of a used car would thus benefit a car dealer in two ways. The first way is that it would allow the car dealer to make a competitive offer when purchasing a used car quickly, and the second way is that it would enable the car dealer to set an optimal selling price for their used car inventory.

This study aimed to determine the value of a vehicle based on predictive modeling of key features to identify vehicles for sale that are either a bargain and could be sold for a profit or overpriced and should be avoided.

## Methods

### Data Exploration

The data that was used for this study is from the “100,000 UK Used Car Data” dataset provided by Kaggle [2]. The dataset contains ten different variables that describe the vehicle: make, model, year, price, transmission, mileage, fueltype, tax, mpg, and enginesize. The data looks at 100,000 cars of ten different car manufacturers from 1997 to 2020. Data from seven different vehicle manufacturers were used: Mercedes, Audi, BMW, Ford, Hyundai, Volkswagen, and Toyota. The original focus was solely on luxury vehicles, but as part of the data exploration, this focus was adjusted to included non-luxury vehicles.

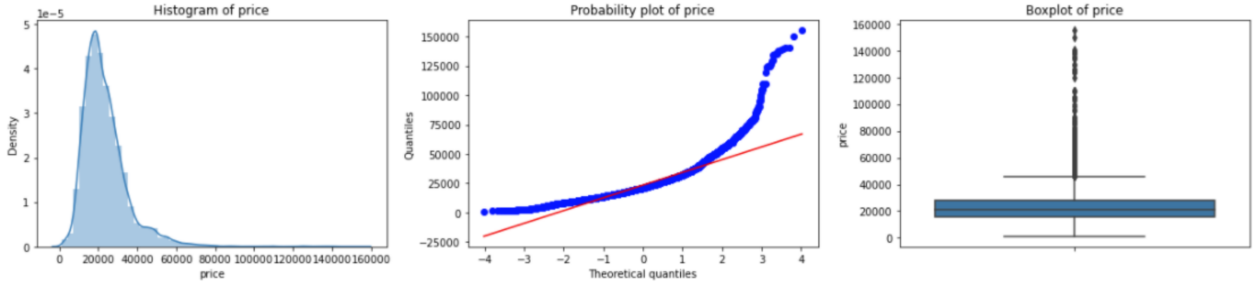
In order to better understand the variables and how they impact the prediction of price, exploratory data analysis (EDA) was done on each dataset individually prior to joining as one dataset. The dataset was split into training and testing sets (70% training) prior to performing EDA to avoid data snooping of unseen data. After the initial EDA, all functions were applied to the testing set. Several Python packages, including pandas, seaborn, matplotlib,numpy, sklearn and scipy, were used to understand relationships and patterns in the data that better informed the modeling process.

### Data Preparation

Combining the seven datasets into a single dataset was simple and direct as all datasets had the same features and vehicle column manufacturer was added. Price, the target variable, was observed as skewed, therefore requiring transformation prior to modeling. This was done within the setup function using the PyCaret library. As seen in Figure 1, there was a wide range in price of vehicles in the dataset, ranging from $675 to $154,998, with an average value of $18,096.

There were several variables with outliers, such as fuel type, where only four out of 55,000 cars had electric engines. These records were removed to avoid skewing the data. Additionally, records where transmission was listed as “other” were removed from the data set, as well as records where fuel type was listed as other or electric. Some records indicated an engine size of zero, where it is more likely the value was omitted, as an engine size of zero is not possible.  The median of engine size was used to replace values that were zero. Vehicles with mileage less than 15 mpg were replaced with the median mpg value from vehicles with mileage greater than 15 mpg. Interestingly, all the low mileage vehicles were from years 2019 and 2020.  Vehicles prior to 1980 and after 2021 were removed from the dataset, as well as vehicles with mileage greater than 200,000 and mpg greater than 300 mpg.  Vehicles were categorized based on the manufacturer as either luxury or standard.

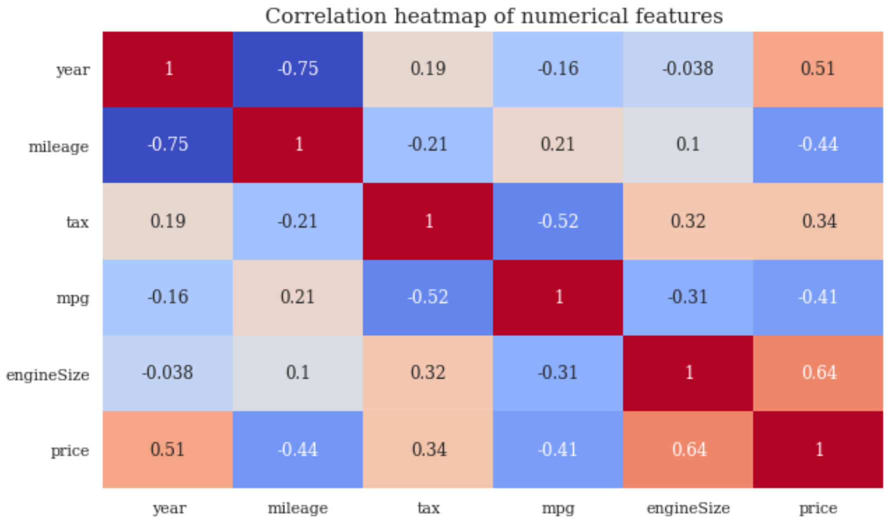
*Figure 1:  EDA of Price Variable*



A correlation matrix for the combined dataset showed low to moderate correlations between the variables. Therefore we included all variables in modeling.  Any multicollinearity observed would need to be removed before using multiple regression or multivariate regression as it would be redundant in the analysis.  Multicollinearity present in the data distorts the results of individual predictor variables.[3]

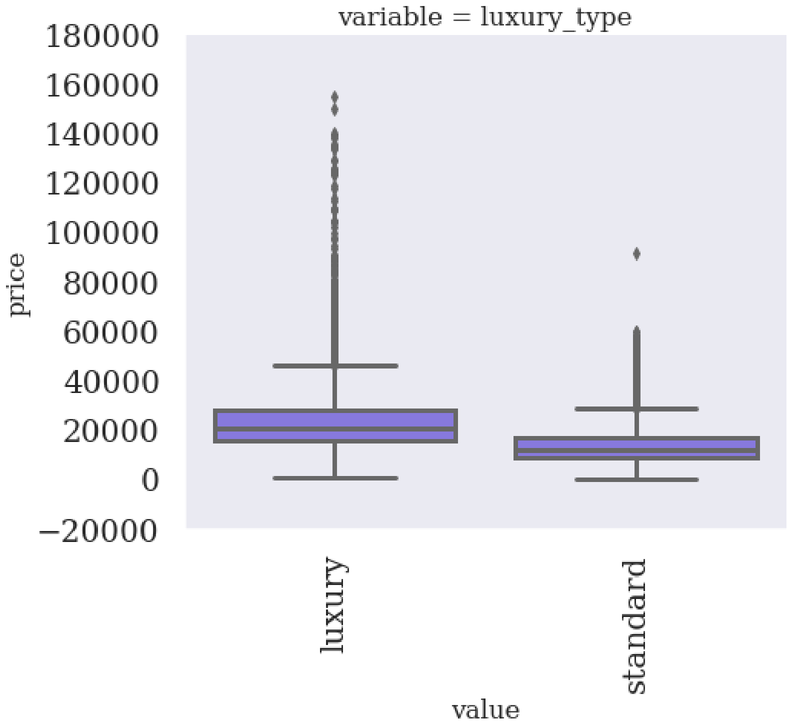
In the correlation heatmap below, year and engineSize were shown to have a moderately strong positive correlation with price.  This indicates that newer vehicles with larger engines have higher prices.  Additionally, mpg and mileage variables have a moderately strong negative correlation with price. These correlations suggest that lower mileage cars tend to have higher prices.

*Figure 2: Correlation Heatmap*



Vehicles were categorized as standard and luxury, and the boxplot below shows the difference in price, thus indicating that this variable will be useful at estimating price in modeling (Figure 3).

*Figure 3: Boxplot of Luxury Type vs Price*



## Modeling

Using the PyCaret library, several regression models were fit to the data using 5-fold cross-validation. The results were evaluated based on appropriate metrics, and we selected five regression models to examine further. $R^2$ was used to determine how much of the variation in price could be explained by the independent variables.

## Results

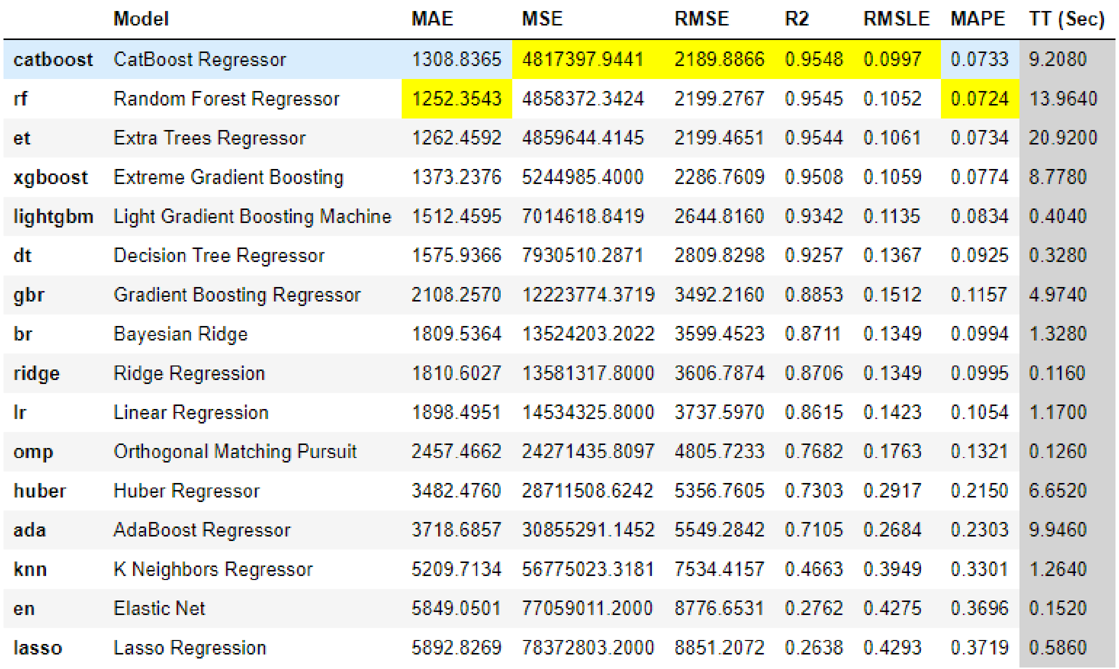
### Baseline Model Performance

Linear regression makes predictions for continuous, real variables. The algorithm shows the linear relationship between a dependent and multiple independent variables. Multiple linear regression is the appropriate approach for the data due to the number of independent variables in the data set. The purpose is to find the best fit line where error between predicted and actual values are minimized. Mean Squared Error (MSE) cost function may be used as a hyperparameter to determine the best fitting line. MSE is the average of squared error between predicted values and actual values. Next, gradient descent can be used to minimize the MSE [4].

There are assumptions of linear regression that are worth noting, firstly, that the relationship between dependent and each independent variable is linear. There should be no high correlations between variables, known as multicollinearity. Multicollinearity makes it difficult to analyze the relationship of other independent variables to the dependent or target variable. The correlation heat map shows no multicollinearity in data set. The error terms should be a normal distribution, and this can be checked by using the qq plot where a straight line indicates normal error [4].

Initial baseline regression models were created using PyCaret.[5]. Regression is a supervised machine learning technique that can be used to estimate the relationship between the target variable (price) and the independent variables. The initial baseline models used the original merged datasets prior to cleaning and EDA, to determine the improvements made through preprocessing and feature engineering. Eighteen different models were evaluated and ranked by metric performance, using the default PyCaret settings. As seen in Figure 4, the top five performing models were CatBoost Regressor, Random Forest Regressor, Extra Trees Regressor, Extreme Gradient Boosting, and Light Gradient Boosting Machine. While Linear Regression was not one of the top performing models, it was included as reference.

*Figure 4: Baseline Regression Modeling Results*

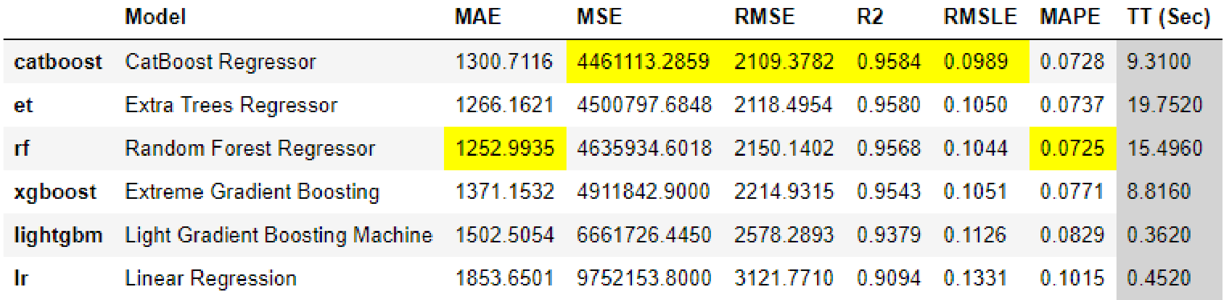


| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **catboost** | CatBoost Regressor | 1308.8365 | 4817397.9441 | 2189.8866 | 0.9548 | 0.0997 | 0.0733 | 9.2080 |
| **rf** | Random Forest Regressor | 1252.3543 | 4858372.3424 | 2199.2767 | 0.9545 | 0.1052 | 0.0724 | 13.9640 |
| **et** | Extra Trees Regressor | 1262.4592 | 4859644.4145 | 2199.4651 | 0.9544 | 0.1061 | 0.0734 | 20.9200 |
| **xgboost** | Extreme Gradient Boosting | 1373.2376 | 5244985.4000 | 2286.7609 | 0.9508 | 0.1059 | 0.0774 | 8.7780 |
| **lightgbm** | Light Gradient Boosting Machine | 1512.4595 | 7014618.8419 | 2644.8160 | 0.9342 | 0.1135 | 0.0834 | 0.4040 |
| **dt** | Decision Tree Regressor | 1575.9366 | 7930510.2871 | 2809.8298 | 0.9257 | 0.1367 | 0.0925 | 0.3280 |
| **gbr** | Gradient Boosting Regressor | 2108.2570 | 12223774.3719 | 3492.2160 | 0.8853 | 0.1512 | 0.1157 | 4.9740 |
| **br** | Bayesian Ridge | 1809.5364 | 13524203.2022 | 3599.4523 | 0.8711 | 0.1349 | 0.0994 | 1.3280 |
| **ridge** | Ridge Regression | 1810.6027 | 13581317.8000 | 3606.7874 | 0.8706 | 0.1349 | 0.0995 | 0.1160 |
| **lr** | Linear Regression | 1898.4951 | 14534325.8000 | 3737.5970 | 0.8615 | 0.1423 | 0.1054 | 1.1700 |
| **omp** | Orthogonal Matching Pursuit | 2457.4662 | 24271435.8097 | 4805.7233 | 0.7682 | 0.1763 | 0.1321 | 0.1260 |
| **huber** | Huber Regressor | 3482.4760 | 28711508.6242 | 5356.7605 | 0.7303 | 0.2917 | 0.2150 | 6.6520 |
| **ada** | AdaBoost Regressor | 3718.6857 | 30855291.1452 | 5549.2842 | 0.7105 | 0.2684 | 0.2303 | 9.9460 |
| **knn** | K Neighbors Regressor | 5209.7134 | 56775023.3181 | 7534.4157 | 0.4663 | 0.3949 | 0.3301 | 1.2640 |
| **en** | Elastic Net | 5849.0501 | 77059011.2000 | 8776.6531 | 0.2762 | 0.4275 | 0.3696 | 0.1520 |
| **lasso** | Lasso Regression | 5892.8269 | 78372803.2000 | 8851.2072 | 0.2638 | 0.4293 | 0.3719 | 0.5860 |

## Baseline Modeling after Preprocessing

Using PyCaret, the top five performing models were fit using 5-fold cross-validation to the cleaned and preprocessed data. For comparison, a linear regression model was also fit to the same data (Figure 5). As previously noted, the target variable, Price, was transformed as part of the PyCaret setup. Various improvements in $R^2$ can be observed, with linear regression $R^2$ increasing from 0.8615 to 0.9094. These improvements are solely based on the preprocessing steps taken during EDA, as no new variables were added.

*Figure 5: Baseline Modeling Results*



| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **catboost** | CatBoost Regressor | 1300.7116 | 4461113.2859 | 2109.3782 | 0.9584 | 0.0989 | 0.0728 | 9.3100 |
| **et** | Extra Trees Regressor | 1266.1621 | 4500797.6848 | 2118.4954 | 0.9580 | 0.1050 | 0.0737 | 19.7520 |
| **rf** | Random Forest Regressor | 1252.9935 | 4635934.6018 | 2150.1402 | 0.9568 | 0.1044 | 0.0725 | 15.4960 |
| **xgboost** | Extreme Gradient Boosting | 1371.1532 | 4911842.9000 | 2214.9315 | 0.9543 | 0.1051 | 0.0771 | 8.8160 |
| **lightgbm** | Light Gradient Boosting Machine | 1502.5054 | 6661726.4450 | 2578.2893 | 0.9379 | 0.1126 | 0.0829 | 0.3620 |
| **lr** | Linear Regression | 1853.6501 | 9752153.8000 | 3121.7710 | 0.9094 | 0.1331 | 0.1015 | 0.4520 |

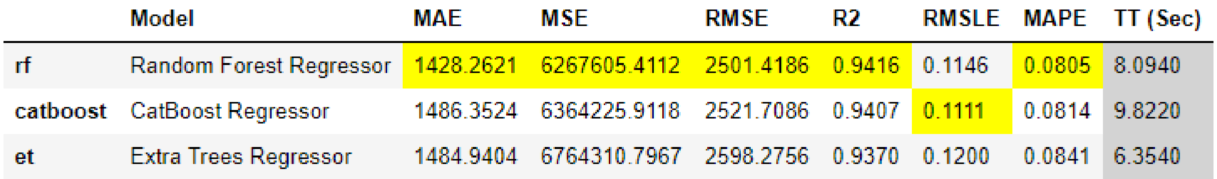
## Feature Engineering

The following features were engineered from the raw data and added to the cleaned and preprocessed dataset: mpg_divide_engineSize, engingeSize_divide_vehicle_age, engineSize_multiply_vehicle_age, mileage_divide_vehicle_age. Since vehicle age is correlated to vehicle year, the vehicle year was excluded going forward to avoid multicollinearity.

Additional data was collected on the average reliability rating of each vehicle manufacturer.[7] The average reliability rating is based on several features of historical data on vehicles, including repair costs, count of failures by make and average age and mileage. The higher the reliability rating, the lower score, as more breakdowns and issues leads to a higher score. We included the variable because we believe that consumers factor in reliability of car prior to purchasing a vehicle.

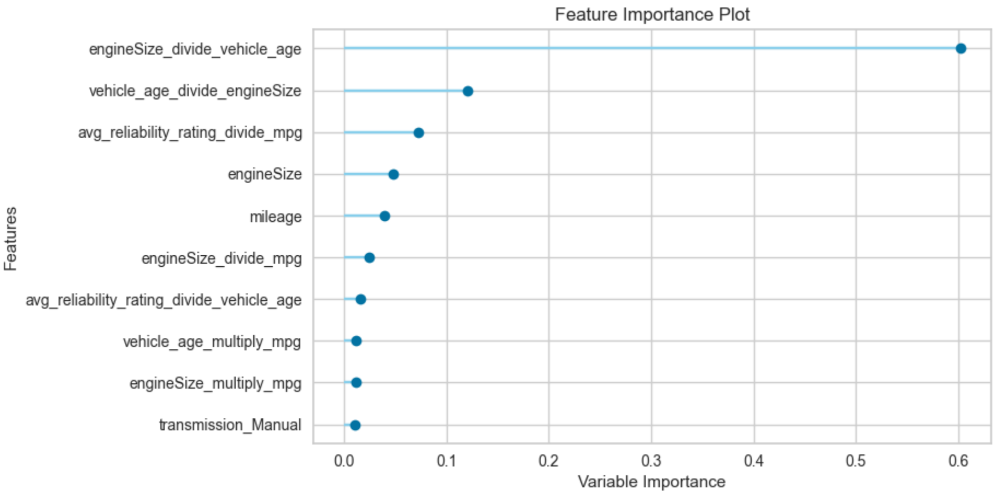
While the vehicle age and reliability rating features were not engineered features, they were used in feature engineering. To engineer new features, the feature interaction and feature ratio parameters were used as part of the PyCaret setup, and three models were created to determine feature importance. Feature interaction is used to create new features by multiplying two variables, while feature ratio is used to create features by calculating the ratios of existing features. PyCaret only includes the engineered features exceeding a given threshold of importance and drops the remaining features before further processing. To avoid exploding the feature space, we excluded the model variable during feature engineering. Figure 6 shows the model results using feature engineering. All regressor performances decreased, but this was likely due to the model variable being excluded from feature engineering. Figure 7 shows the feature importance results using random forest regressor.

*Figure 6: Feature Engineering Model Results*



| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **rf** | Random Forest Regressor | 1428.2621 | 6267605.4112 | 2501.4186 | 0.9416 | 0.1146 | 0.0805 | 8.0940 |
| **catboost** | CatBoost Regressor | 1486.3524 | 6364225.9118 | 2521.7086 | 0.9407 | 0.1111 | 0.0814 | 9.8220 |
| **et** | Extra Trees Regressor | 1484.9404 | 6764310.7967 | 2598.2756 | 0.9370 | 0.1200 | 0.0841 | 6.3540 |

*Figure 7: Feature Importance Plot of Random Forest Regressor*



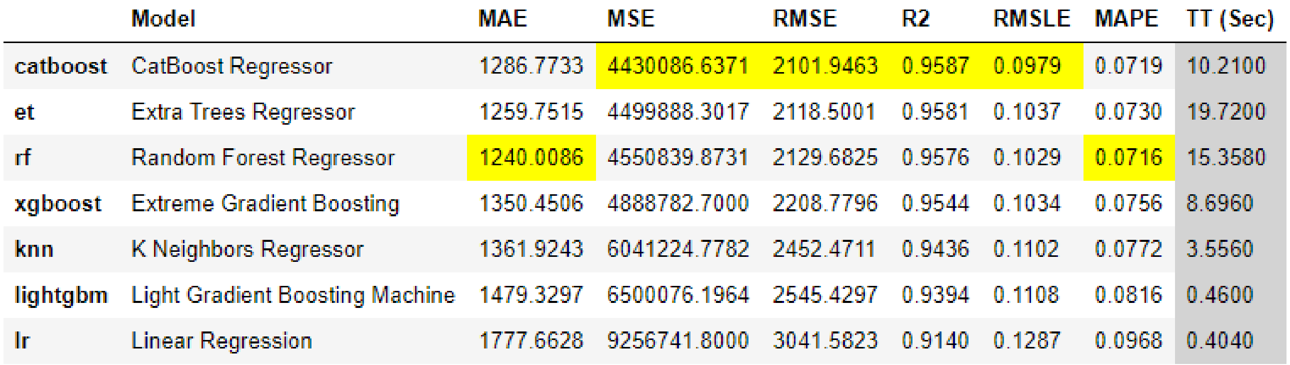
### Feature Selection

Review of the feature importance plots for the top three models shows that several features have high importance in multiple plots. As a result of feature engineering, we added four new features: engineSize_divide_vehicle_age, avg_reliability_rating_divide_vehicle_age, mpg_multiply_vehicle_age, and mpg_divide_engineSize.

### Updated Model Performance

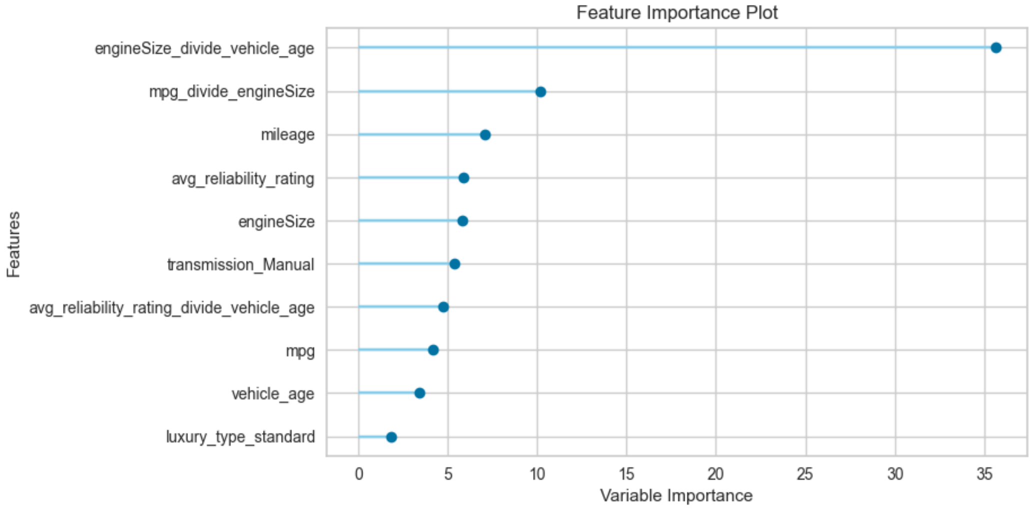
The same top five models, plus linear regression, were again fit to the cleaned and preprocessed data using 5-fold cross-validation, but this time the newly engineered features were included. Additionally, we also fit a KNN model to the same data. The data was normalized using a robust scalar within the PyCaret setup. The results in Figure 8 show increased performance across almost all models. The most noteworthy increase in performance, likely due to normalizing the data, can be observed for KNN where $R^2$ increased from 0.4663 in the baseline model to 0.9436, to become one of the top five performing models. Additional feature importance plots were created using the updated model, and three of the top four most important features were observed to be engineered features (Figure 9).

*Figure 8: Model Performance after Feature Engineering*



| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **catboost** | CatBoost Regressor | 1286.7733 | 4430086.6371 | 2101.9463 | 0.9587 | 0.0979 | 0.0719 | 10.2100 |
| **et** | Extra Trees Regressor | 1259.7515 | 4499888.3017 | 2118.5001 | 0.9581 | 0.1037 | 0.0730 | 19.7200 |
| **rf** | Random Forest Regressor | 1240.0086 | 4550839.8731 | 2129.6825 | 0.9576 | 0.1029 | 0.0716 | 15.3580 |
| **xgboost** | Extreme Gradient Boosting | 1350.4506 | 4888782.7000 | 2208.7796 | 0.9544 | 0.1034 | 0.0756 | 8.6960 |
| **knn** | K Neighbors Regressor | 1361.9243 | 6041224.7782 | 2452.4711 | 0.9436 | 0.1102 | 0.0772 | 3.5560 |
| **lightgbm** | Light Gradient Boosting Machine | 1479.3297 | 6500076.1964 | 2545.4297 | 0.9394 | 0.1108 | 0.0816 | 0.4600 |
| **lr** | Linear Regression | 1777.6628 | 9256741.8000 | 3041.5823 | 0.9140 | 0.1287 | 0.0968 | 0.4040 |

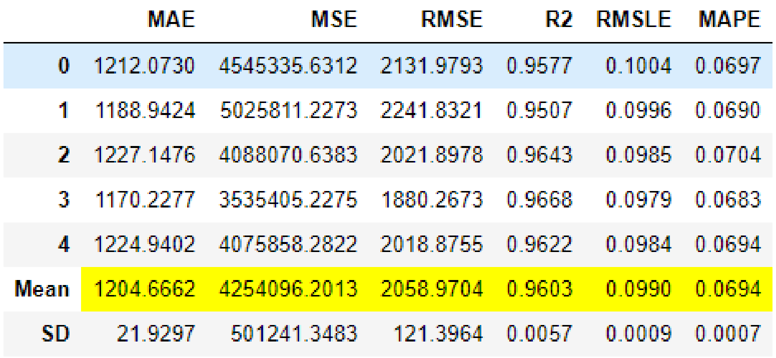
*Figure 9:  Feature Importance Plot of Catboost Regressor*



## Ensemble Methods

Ensemble models in machine learning consist of combing the predictions from multiple models to improve the overall performance.  PyCaret offers various ensemble methods, including, bagging and blending, which were explored further.  Bagging was done on the extra trees regressor and slight improvement in performance can be observed, as shown in Figure 10.

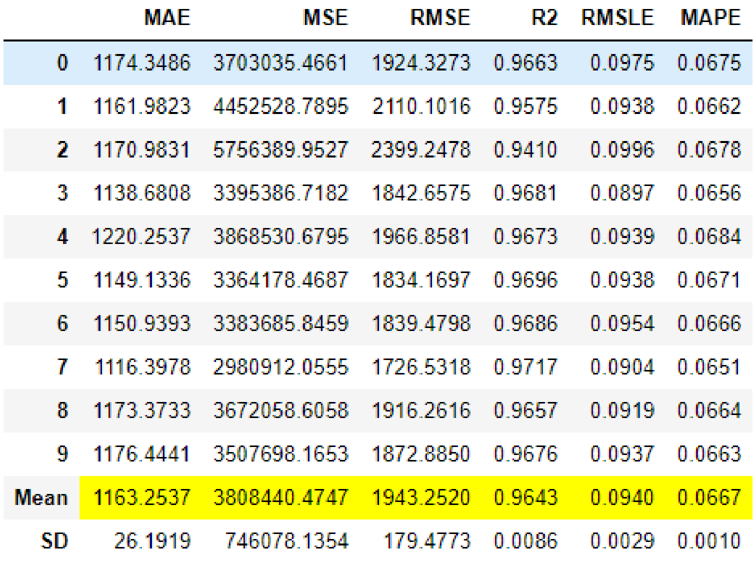
*Figure 10:  Bagging Extra Trees Regressor Results*



| | MAE | MSE | RMSE | R2 | RMSLE | MAPE |
|---|---|---|---|---|---|---|
| **0** | 1212.0730 | 4545335.6312 | 2131.9793 | 0.9577 | 0.1004 | 0.0697 |
| **1** | 1188.9424 | 5025811.2273 | 2241.8321 | 0.9507 | 0.0996 | 0.0690 |
| **2** | 1227.1476 | 4088070.6383 | 2021.8978 | 0.9643 | 0.0985 | 0.0704 |
| **3** | 1170.2277 | 3535405.2275 | 1880.2673 | 0.9668 | 0.0979 | 0.0683 |
| **4** | 1224.9402 | 4075858.2822 | 2018.8755 | 0.9622 | 0.0984 | 0.0694 |
| **Mean** | 1204.6662 | 4254096.2013 | 2058.9704 | 0.9603 | 0.0990 | 0.0694 |
| **SD** | 21.9297 | 501241.3483 | 121.3964 | 0.0057 | 0.0009 | 0.0007 |

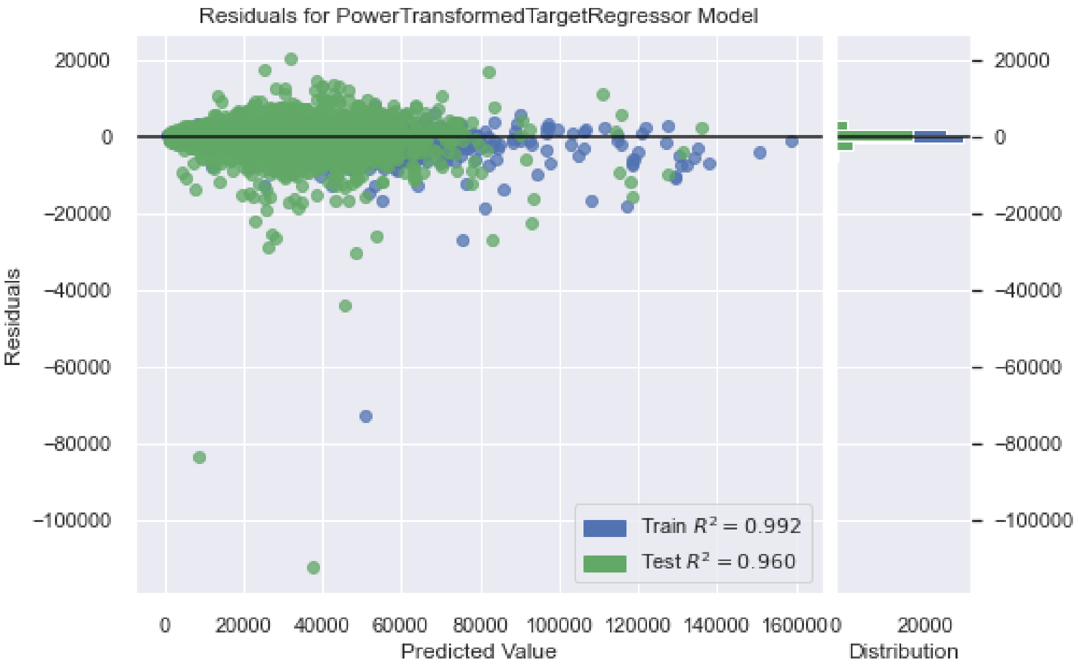
Based on the model results from PyCaret, three models were selected for blending, catboost, extra trees, and random forest.  The final models were selected based on several of factors, including overall performance and ease of explaining output to the final audience. The overall R² of the blended ensemble is 0.96, which is a slightly better result from the individual models.

*Figure 11: Results from Blending Catboost Regressor, Extra Trees Regressor and Random Forest Regressor*



| | MAE | MSE | RMSE | R2 | RMSLE | MAPE |
|---|---|---|---|---|---|---|
| **0** | 1174.3486 | 3703035.4661 | 1924.3273 | 0.9663 | 0.0975 | 0.0675 |
| **1** | 1161.9823 | 4452528.7895 | 2110.1016 | 0.9575 | 0.0938 | 0.0662 |
| **2** | 1170.9831 | 5756389.9527 | 2399.2478 | 0.9410 | 0.0996 | 0.0678 |
| **3** | 1138.6808 | 3395386.7182 | 1842.6575 | 0.9681 | 0.0897 | 0.0656 |
| **4** | 1220.2537 | 3868530.6795 | 1966.8581 | 0.9673 | 0.0939 | 0.0684 |
| **5** | 1149.1336 | 3364178.4687 | 1834.1697 | 0.9696 | 0.0938 | 0.0671 |
| **6** | 1150.9393 | 3383685.8459 | 1839.4798 | 0.9686 | 0.0954 | 0.0666 |
| **7** | 1116.3978 | 2980912.0555 | 1726.5318 | 0.9717 | 0.0904 | 0.0651 |
| **8** | 1173.3733 | 3672058.6058 | 1916.2616 | 0.9657 | 0.0919 | 0.0664 |
| **9** | 1176.4441 | 3507698.1653 | 1872.8850 | 0.9676 | 0.0937 | 0.0663 |
| **Mean** | 1163.2537 | 3808440.4747 | 1943.2520 | 0.9643 | 0.0940 | 0.0667 |
| **SD** | 26.1919 | 746078.1354 | 179.4773 | 0.0086 | 0.0029 | 0.0010 |

*Figure 12: Comparison of Training vs Test Data for Blended Ensemble Model*



## Discussion/Conclusion

Overall, all models performed quite well, as all models explained 95% or greater of the variance of used car prices in our dataset. Because of this, there were only minor improvements from baseline model using featuring engineering and ensemble models. Across all individual modeling techniques, Catboost performed the best at 0.9587, but using a blending ensemble technique led to best results (0.9643).

*Figure 13: Comparison of Training vs Test Data for Blended Ensemble Model*

| **Model Type** | **Model Name** | **$R^2$** | **RMSLE** |
|---|---|---|---|
| Baseline | CatBoost Regressor | 0.9548 | 0.0997 |
| Baseline | Linear Regression | 0.8629 | 0.1036 |
| Enhanced | CatBoost Regressor | 0.9587 | 0.0979 |
| Enhanced | Linear Regression | 0.9129 | 0.1292 |
| Ensemble | Boosting | 0.9592 | 0.1022 |
| Ensemble | Bagging | 0.9603 | 0.099 |
| Ensemble | Blending | 0.9643 | 0.094 |

While we are happy with results, there are also several areas worth discussing that impacted the underlying data and model development. We noticed in the EDA that there was a sharp drop in mpg for cars built in 2018-2020 compared to the upward trend from previous years. We researched the issue and while there were updated regulations regarding the reporting of gas mileage start around that time due to lawsuits, we could not definitively determine why the drop. This may have impacted model performance if newer used cars mpg were inaccurate in the dataset. While we worked with missing data and removed data that was obviously wrong (eg a car built in 2060), there is gray area dealing with data that doesn't seem to match real-life performance. If we had more time, we likely would have worked to derive average mpg for each vehicle by year from the manufacturer's websites and used that instead of using the mpg data from the dataset. However, the mpg feature did not come up as an importance feature, so it likely had a small impact on the analysis. Something important to note in future analysis, is that it is important to not only clean the data, but also make sure the data you are working with is valid.

Regarding the modeling process itself, while we are pleased with the overall performance of our models, we must be careful with comparisons between the different models as well. As we know, R squared will only increase as more variables are added[10]. So, while adding new features in our dataset helped improve our model, the difference in R squared values may have been inflated simply because we added more variables to the model with feature engineering. In the future, we would look at multiple comparison metrics (including adjusted which accounts for the issue above) to make sure that we are accounting for difference in variables including in each modeling step.

While our analysis was focused on maximizing the $R^2$ values, we could have also chosen a different metric to optimize prior to model deployment. While the ensemble blending model had the best performance, it was built by combining multiple models together, which makes it harder to understand the relationships between the variables and the prediction. While linear regression was not the top performing model, it may be a preferred model for deployment because it still had an $R^2$ value of .91 after feature engineering and it is the easiest to explain how the prediction is calculated. The small improvement in $R^2$ value in the ensemble model may be outweighed by the interpretable results of a linear regression model. It was also observed that for the feature importance for the enhanced linear regression, the top two variables were ones we derived or added to the analysis (luxury type and reliability rating), showing that our EDA and feature engineering allowed us to create a more powerful model than by simply using the baseline data set (.86 $R^2$ vs .91 $R^2$).

## Acknowledgments

We would like to thank Professor Werner for his guidance and advice while working through our project for this class. We would also like to thank our families for supporting our efforts throughout this class and this program that allows us to develop our skills in Data Science.

## References

[1] Rosenbaum, Eric. (2020, October 16). The used car boom is one of the hottest, and trickiest, coronavirus markets for consumers. Retrieved December 13, 2020, from https://www.cnbc.com/2020/10/15/used-car-boom-is-one-of-hottest-coronavirus-markets-for-consumers.html

[2] Aditya. (2020, July 04). 100,000 UK Used Car Data set. Retrieved December 13, 2020, from https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

[3] Multicollinearity. (2020, December 26). Retrieved January 16, 2021, from https://en.wikipedia.org/wiki/Multicollinearity

[4] Linear Regression in Machine learning - Javatpoint. (n.d.). Retrieved January 10, 2021, from https://www.javatpoint.com/linear-regression-in-machine-learning#:~:text=Linear%20Regression%20in%20Machine%20Learning%201%20Types%20of,the%20set%20of%20observations.%20...%20More%20items...%20

[5] https://pycaret.org/

[6] Pulkit Sharma. (2020, December 07). Model deployment Using STREAMLIT: DEPLOY Ml models using Streamlit. Retrieved February 21, 2021, from

https://www.analyticsvidhya.com/blog/2020/12/deploying-machine-learning-models-using-streamlit-an-introductory-guide-to-model-deployment/

[7] How reliable is your car or the car you're about to buy? https://www.reliabilityindex.com/

[8] Mahendru, K. (2019, August 13). Measuring the goodness of fit: $R^2$ versus adjusted $r^2$. from https://medium.com/analytics-vidhya/measuring-the-goodness-of-fit-r%C2%B2-versus-adjusted-r%C2%B2-1e8ed0b5784a

[9] Saxena, S. (2020, February 13). Rmse vs rmlse - what's the difference? When should you use them?, from https://medium.com/analytics-vidhya/root-mean-square-log-error-rmse-vs-rmlse-935c6cc1802a

[10] Fernando, J. (2020). R-Squared Definition. Retrieved from: https://www.investopedia.com/terms/r/r-squared.asp