CarMax Trade-In Analysis

–

ORIE 4741: Final Project

Artem Ezhov

May 12, 2023

Abstract

Providing personalized customer experience is essential for car dealerships because it can help build long-term relationships with customers and increase sales. In today's highly competitive automotive industry, it's not enough to simply sell cars; dealerships must differentiate themselves by creating a unique and personalized experience for customers. By leveraging customer data and insights, dealerships can tailor their approach to meet each customer's unique needs and expectations.

CarMax, a renowned American used-car retailer, operates in more than 255 stores across the country. Customers who purchase a car from CarMax often use their current vehicle as a trade-in, for which they receive an appraisal offer. This project aims to analyze a sample of CarMax appraisal data to understand the relationship between a customer's appraised vehicle and the vehicle they purchase. By doing so, we hope to identify insights that can help CarMax improve its business operations and provide customers with a more personalized shopping experience.

Contents

1 Exploratory Data Analysis

1.1 Data Description

The data set we shall use for this project was published as a part of CarMax Analytics Showcase Winter 2023. It is a sample of 200,000 customers who have purchased a car from CarMax and their accompanying appraisal details. The data set includes vehicle attributes for purchased and appraised cars, including make/model/trim, mileage, appraisal, and purchase value. Appraised vehicle data can be identified by the suffix '_appraisal'. 'Among 30 features, there are 17 strings, 12 numeric, and 1 integer variables. With this dataset, we would like to examine how various features influence the price of a customer's purchased vehicle. We will try to capture such relationships by building different models after some appropriate data preprocessing.

1.2 Data Visualization

1.2.1 Variable Correlations

To determine the relatively important aspects of determining the price of a newly purchased vehicle, we created a correlation plot for numerical features.

	price	appraisal_offer	online_appraisal_flag	model_year_appraisal	mileage_appraisal
price	1.000	0.390	0.037	0.284	-0.246
appraisal_offer	0.390	1.000	0.175	0.723	-0.737
online_appraisal_flag	0.037	0.175	1.000	0.172	-0.167
model_year_appraisal	0.284	0.723	0.172	1.000	-0.741
mileage_appraisal	-0.246	-0.737	-0.167	-0.741	1.000
engine_appraisal	0.232	0.178	-0.027	-0.175	0.151
cylinders_appraisal	0.217	0.154	-0.031	-0.205	0.158
mpg_city_appraisal	-0.182	-0.091	0.033	0.192	-0.160
mpg_highway_appraisal	-0.203	-0.124	0.034	0.236	-0.183
horsepower_appraisal	0.350	0.415	0.025	0.105	-0.069

Figure 1: Correlation Plot

As we see in Figure 1, the price of the customer's purchased vehicle is slightly correlated with the appraisal_offer, the model year and horsepower of the customer's appraised vehicle. This inference is in line with the reality as the customer who received a better appraisal offer would be more likely to purchase a more expensive vehicle. Similarly, if a customer trades in a newer model of the car, the chances are that the newly purchased vehicle would be a newer model, which would be reflected in its price. Some numerical features seem to have little correlation with the price so it's appropriate to use regularization or variable selection while training models.

1.2.2 Prices of Purchased Vehicles

The distribution of CarMax purchased vehicles prices is presented in Figure 2, which indicates a right-skewed distribution. To address this issue, we applied a log transformation to the target data, as we will explain in more detail in a later section of data processing.

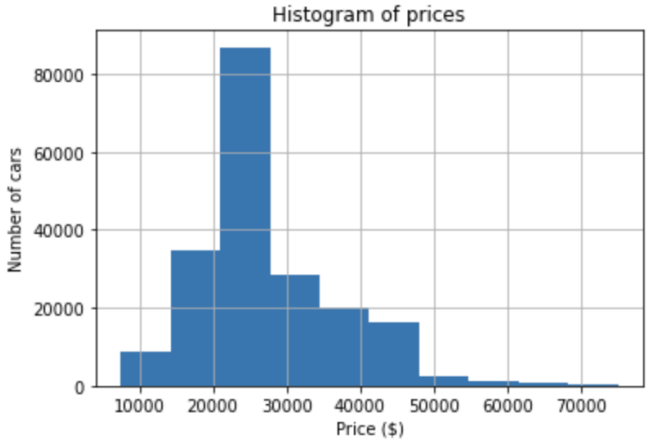

Figure 2. CarMax Prices Histogram

As we have previously identified the correlation between the price and appraisal offer, we plotted the distributions of purchased vehicle prices according to each appraisal offer category. As we can see from Figure 3, the average price of the highest appraisal offer has the highest corresponding price of the purchased vehicle, and vice versa. These observations indicate that the appraisal offer information could be used to predict the price of a vehicle purchased from CarMax.

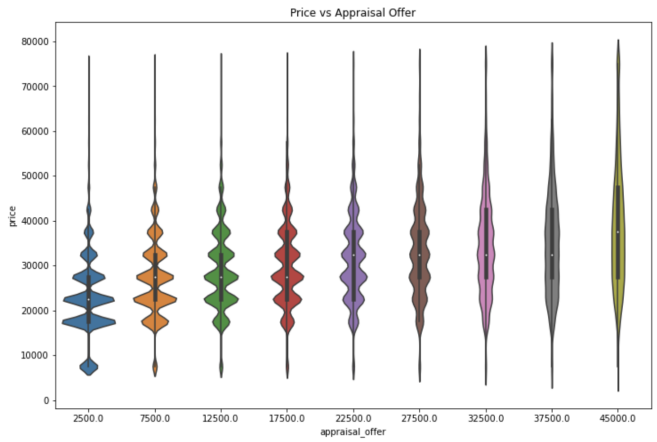

Figure 3. Price vs. Appraisal Offer

1.3 Data Preprocessing

1.3.1 Outliers

Although the conventional definition of an outlier is a data point that falls more than 1.5 times the interquartile range below the first quartile or above the third quartile, we did not apply this definition in our analysis due to the unique nature of the data. We rather used common sense to eliminate outliers, such as cases when people trade in $75k cars for $5k cars.

1.3.2 Nominal Values

The columns “make_appraisal” and “model_appraisal” contain nominal values. Since “model_appraisal” specifies the model of a vehicle manufactured by “make_appraisal” company, the two columns contain overlapping information. Additionally, each make can have multiple models which align to the numbers in the model column, however each make has a unique process for designating these values, so across makes the model numbers aren’t comparable. Hence, it has been decided to drop the “model_appraisal” column and leave only “make_appraisal” to avoid correlation-related problems while keeping necessary information about the manufacturer. Since there is no ordinal relationship between different manufacturers, we applied one-hot encoding on the remaining column to facilitate our model building and predictions. The columns, “trim_descrip_appraisal” and “body_appraisal”, containing few categories have been also encoded using one-hot encoding.

The dataset also contains information about the colors of the appraised vehicles. The “color_appraisal” column has been encoded by converting every color to a RGB representation, producing 3 new columns. However, the careful examination of the correlation plot concluded that there is no correlation between colors of appraised vehicles and the prices of a purchased vehicle.

1.3.3 Continuous values

To analyze the relationship between engine size and car price, we converted the “engine_appraisal” column from liter to numeric representation, e.g. 4.0L has been converted to 4.0 The ‘mileage_appraisal’ column has been provided in the following format “10k to 20k miles” in increments of 10k, and we converted it to an integer representation of the average of two numbers, 15000 in the example provided. This conversion process was essential for our analysis and ensured that we could make meaningful and accurate conclusions.

1.3.4 NA Values

Due to high integrity, the NA values only appeared in very few columns, as in Figure 4.

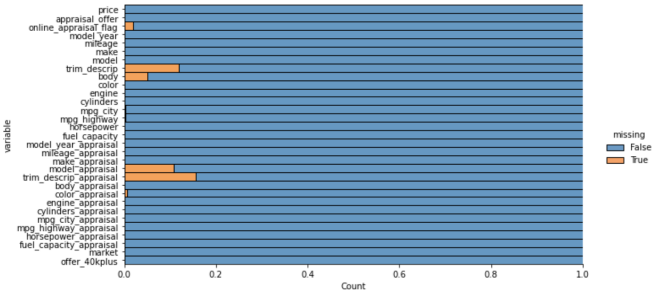

Figure 4. Missing Values

The missing values in the setup of our problem do not convey any information, but rather indicate that the data wasn't recorded/available. For "trim_descip_appraisal", the null values have been filled with most frequent values, as the distribution of different trim descriptions were proportional across different prices.

1.3.4 Response Transformation

Initially, the price of the customer's purchased vehicle has been represented as string type (e.g. "$20k to $25k") in $5k increments up to $70k. The price has been converted to an average value of two bounds of the range, and for extreme values, such as "70k+", an additional binary column has been added to capture the high price of the purchased vehicle. According to the histogram of our price range, the target variable is right-skewed with a long tail to the right. This is consistent with our intuition that most people are going for fair-priced car options rather than premium or luxurious vehicles. To mitigate the impact of expensive car listings and increase the stability of our data's variance, we can transform the distribution of prices by logarithmically scaling their values. This approach will enhance the accuracy of our regression models in subsequent phases.

2 Regression Modeling

2.1 Model Evaluation

We opted to use the quadratic loss function for model evaluation over other loss functions, as we are dealing with a continuous target variable. We would like to avoid large prediction errors, as it would hurt the personalized shopping experience for a customer. In this case, quadratic loss is mathematically more tractable and suitable for the purposes of our analysis.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

In particular, we used Root Mean Squared Error to evaluate our model, which compared the logarithm of predicted and actual prices. As discussed previously, taking the logarithm ensures that errors in predicting both high and low prices had an equal impact on the outcome.

2.2 Linear Model

Before fitting linear models, features have been scaled using Robust Scaler, which removes the median and scales the data according to the quantile range. By doing so, we make features robust to outliers.

To avoid the linear dependence between features and non-unique solutions, it is necessary to introduce a regularization term to guarantee the integrity of our linear model.

2.2.1 Lasso Regression

$$minimize \sum_{i=1}^{n}(y_i - w^\intercal x_i)^2 + \lambda \sum_{i=1}^{n} w$$

To ensure a unique solution, we introduced an L1 regularizer to the least square optimization problem. Using the Lasso model as our baseline allowed us to establish a performance benchmark for our dataset and gain insights into the significance of the variables in determining the prices of customers' purchased vehicles.

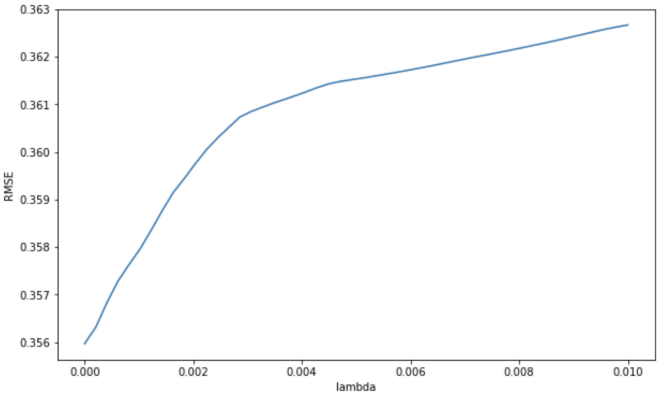

Figure 5. 5-fold CV for Lasso Regression

Next, we utilized a 5-fold cross-validation to identify the optimal value of λ that would minimize RMSE. With optimal λ being 1e-10, we managed to obtain a linear model with a test RMSE of 0.35598 and train RMSE of 0.35585. To determine whether our model might be underfitting and whether it is appropriate to use hard variable selection, we try other regularizers.

2.2.2 Ridge Regression

$$minimize \sum_{i=1}^{n}(y_i - w^\intercal x_i)^2 + \lambda \sum_{i=1}^{n} w^2$$

In order to maintain all of our features while also ensuring a unique solution, we opted to substitute the L1 regularizer with a quadratic regularizer.

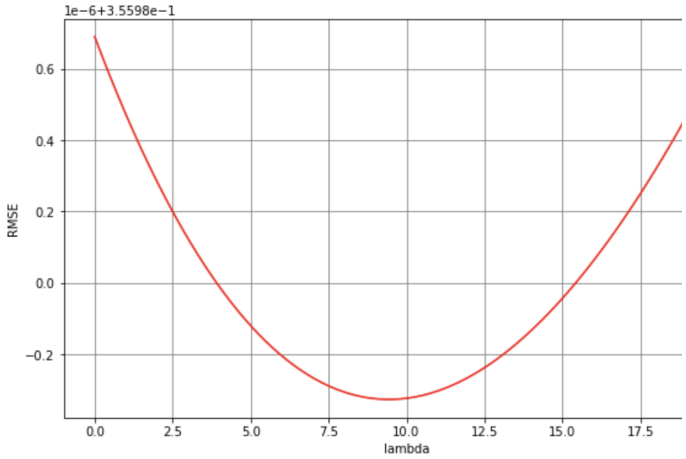

Figure 6. 5-fold CV for Ridge Regression

Using 5-fold cross-validation as previously, we identified the optimal λ to be equal to 9.5, as shown in Figure 6. The RMSE on the test and train data turned out to be 0.35597 and 0.35585, accordingly. It is evident that there is not much improvement on test scores when comparing Ridge Regression to Lasso Regression.

Generally, linear models are not great predictors as they produce high errors if the relationship between the independent variables and the dependent variable is not linear. To solve this problem, we could either gather more data that would be related to the information about newly purchased vehicles or try using more complex regression models on our data. As we are operating with limited resources of finding additional information that could improve our linear models, the next step would be to fit more complex models.

2.3 Tree-Based Models

There is room for improvement as linear models do not capture all possible nonlinear relationships between features and responses. To address this issue and increase prediction accuracy, we will explore more complex tree-based models. Given the size of our data, a single tree is likely to underfit. Therefore, we plan to utilize bagging and boosting techniques on tree-based models to increase flexibility and control variance.

2.3.1 Random Forest Regressor

A Random Forest Regressor is a type of ensemble learning method in which multiple decision trees are trained on different subsets of the training data and the final prediction is made by averaging the outputs of all individual trees. It is a powerful algorithm due to its robustness and ability to handle high-dimensional data.

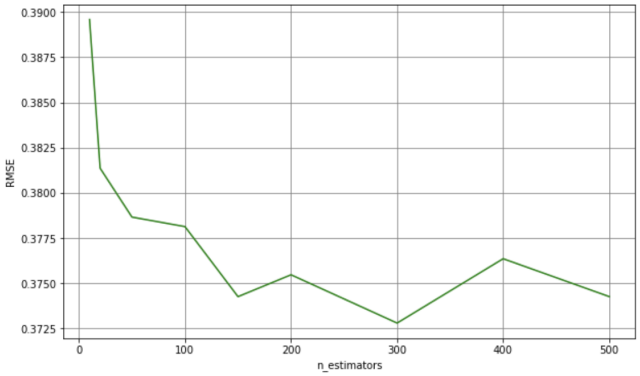

Figure 7. RMSEs for Random Forest with different number of trees

The parameter we have chosen to tune is the number of trees in the forest. In particular, we grid searched over 9 evenly spaced possible n_estimators parameters, ranging from 10 to 500: 10, 20, 50, 100, 150, 200, 300, 400, 500. Figure 7 indicates that we can attain the minimum RMSE of 0.37472 by training 300 distinct trees. However, the outcome did not demonstrate any advancement over previous regularized linear models; instead, it exhibited inferior performance compared to the linear models. Usually, the random forest algorithm is effective on an extensive dataset with a large number of features. So, we conclude that the small number of features in our present dataset may not be able to unleash the full potential of this algorithm.

2.3.2 Gradient Boosting Regressor

Gradient Boosting Regressor is an iterative method that builds decision trees one at a time, where each tree attempts to correct the errors made by the previous one. It works by sequentially fitting decision trees on the residuals of the previous tree, and combining their predictions. Since GBR has a tendency to overfit the training data, especially when the number of trees is large or when the learning rate is too high, we will try to carefully tune different boosting parameters and keep the learning rate low.

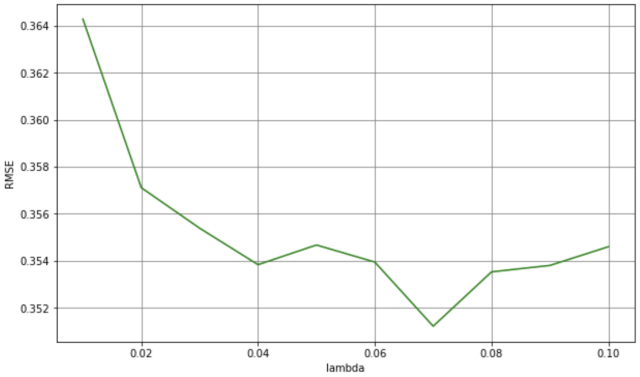

Figure 8. RMSEs for GBR with different learning rates

After training on the dataset, we found the optimal training rate to be 0.7, which produces the cross validation error of 0.3532.In comparison to random forest regressor, gradient boosting provides more flexibility and has a better prediction accuracy.

2.3.3 XGBoost Regressor

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that includes several key enhancements for speed, performan, and scalability. Like gradient boosting, XGBoost works by iteratively adding decision trees to an ensemble model, with each model attempting to correct the errors of the previous trees. XGBoost analyzes the distribution of features across all data points in a leaf and reduces the search space for possible feature splits based on previous information, making it more efficient than gradient boosted trees. With all these advantages, the performance of XGBoost heavily depends on the parameters tuning and does not guarantee a better performance in every case, as the error rate on each model depends heavily on the dataset itself. To stay consistent, we used the same learning rate of 0.7 and after 5-fold cross-validation, XGBoost Regressor produced an RMSE of 0.3520.

2.3.4 Model Selection

According to the table provided, the extreme gradient boosting shows the best performance using RMSE error metric.

Cross-Validation RMSEs for different regressors	
Regressor	RMSE
Lasso	0.35598
Ridge	0.35597
Random Forest	0.37472
Gradient Boosting	0.35320
XGBoost	0.35209

Table 1: Model Selection

3 Conclusion

Predicting the price that a customer might purchase through trade-in after receiving an appraisal offer might help CarMax provide their customers a more personalized shopping experience. In the scope of this project, we tried building a price prediction model based on a dataset provided by CarMax for Winter Analytics Showcase 2023. We first used linear regression and then tried several more decision-based complex models. The gradient boosting regressor turned out to show the best performance on the dataset. Therefore, one should use XGBoost Regressor to get a better reference for a range of prices for a new vehicle based on the customer's appraised vehicle.

4 Limitation and Fairness

4.1 Weapon of Math Destruction

Upon careful examination of our modeling process and outcomes, we are confident that our project does not create a weapon of math destruction. First, the outcomes are clearly stated and easy to measure. We can easily measure the accuracy of our predictions by comparing them to the true prices of customers' purchased vehicles. Secondly, the modeling process is static and unidirectional. We predict prices based on the relevant features of the appraised vehicles, so there is no feedback loop. Thirdly, the conclusions provided by our model can only serve as a reference in the specific scope of the project, and do not have the ability to influence market prices or harm anyone's interests.

4.2 Fairness

When implementing algorithms to analyze data problems, it is important to consider potential biases in the available information that could lead to incorrect results and unintended negative consequences. Therefore, it is essential to evaluate the fairness of our model. We have little insight on how the data was collected by CarMax, so we can not say anything about bias introduced at that step. The data is composed of vehicle attributes visitors appraised and the vehicles they purchased at Carmax - including make/model/trim, mileage, appraisal and purchase value. Most of these values are values assigned by car manufacturers. Since the features of our dataset do not relate to discrimination, such as gender and race of a customer, the topic of fairness is not relevant to the project.

5 Potential Improvement

Most of the values about appraised vehicles are assigned by car manufacturers, and cars are made with similar specifications and capabilities. Hence, we can try treating some of the variables as categorical rather than numerical. Even "price" and "appraisal_offer" can be represented as a range of values, and hence can also be treated as a categorical variable. This approach enables to frame the problem of prediction prices categories as a classification problem, rather than a regression problem.

The dataset also contains information regarding the purchased vehicle itself, which can also be used to predict features of a newly purchased vehicle. This information can be used by CarMax to improve business operations and provide a personalized customer experience.