Used Car Sale Price Prediction Using XGBoost Regressor
DATA 1030 Final Report
Instructor: Andras Zsom

Tongtong Zhao
Github: https://github.com/Christy9615/DATA1030FinalProj

Dec 7, 2021

1. Introduction

As the Covide-19 keeps impacting all kinds of things globally, the used car's price cannot escape from this effect. For example, in China, the price has increased more than 40% compared to two years ago. Thus, the prediction of the price trend of the used car has become a popular topic now. Therefore, this project will benefit the people interested in the topic and the car dealers and customers in that field. The dataset is downloaded from a Machine Learning competition website ---"Tianchi," the competition is still ongoing; only a few EDA has been posted for this dataset, no publication is available yet. Those EDA did missing values replacement as well as visualization. There are two parts of this used car dataset: one is for training and one for testing. The only difference is that the training dataset has the target variable "Price," which does not exist in the test dataset. In addition, there are 30 features plus one target variable in the training dataset, and the test dataset has 30 features same as those in the training dataset. Besides, 150000 data points are stored in the training dataset, 50000 data points are in the test dataset, and all of them are desensitized. The 30 features are: ID, features of a used car(name, fuel type, power, registered date, created date(the date that car has been posted), kilometer that it has been driven, body type, brand, model, gearbox, repaired for damage or not, region code, offer type(request or providing), seller), v0 to v14 (15 features) are embedding vectors that come from car evaluation, customer experience and so on, there is no further description on those anonymous features. Since the project's ultimate goal is predicting the price for the cars in the test dataset, which is considered a quantity, it is a regression problem.

2. Exploratory Data Analysis

In the EDA section, the dataset is observed from the shape, missing value, correlation between features and target variable, data distribution in each numerical and categorical features. The graphs below are generated from this section.

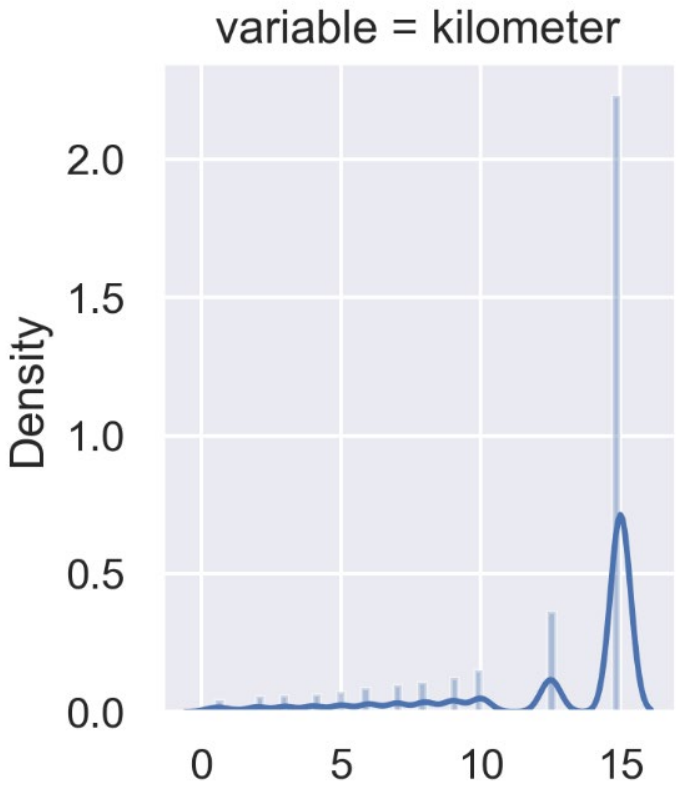

Figure 1 This histogram shows the distribution of used cars' driving distance. From the figure, most of used cars' driving distance are located above 14 kilometers range, which may have a negative correlation to price.

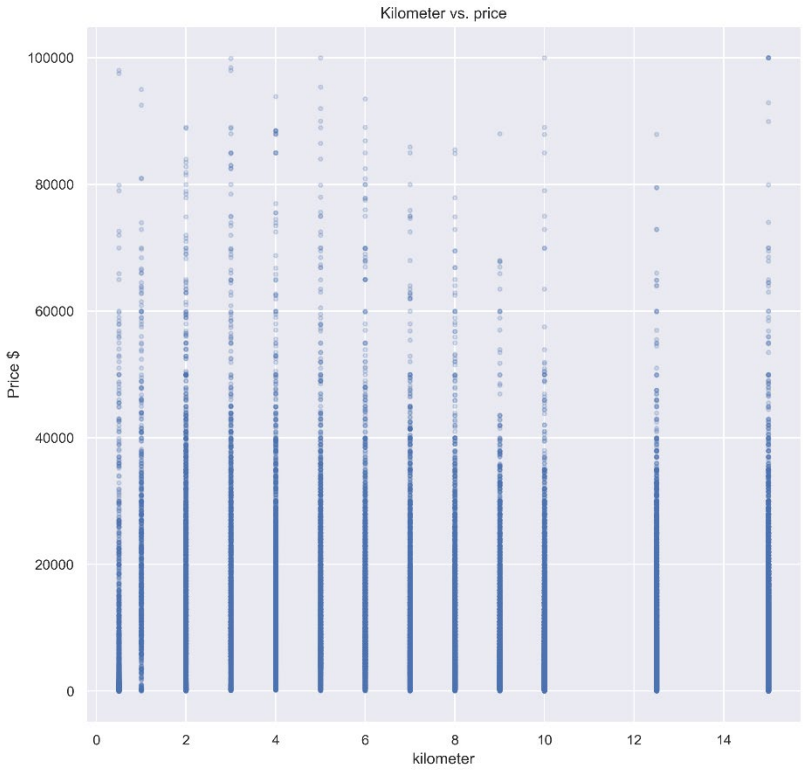

Figure 2 This scatter plot shows relationship between trading price and kilometer. The cars that have a high driving usually has a lower trading price. But there also some cars that have a low range in kilometer get a low value of price. Similarly, there is an outlier in the greater than 140000 meters range, which means that car has a 140k+ driving distance but has a trading value about 100k RMB.

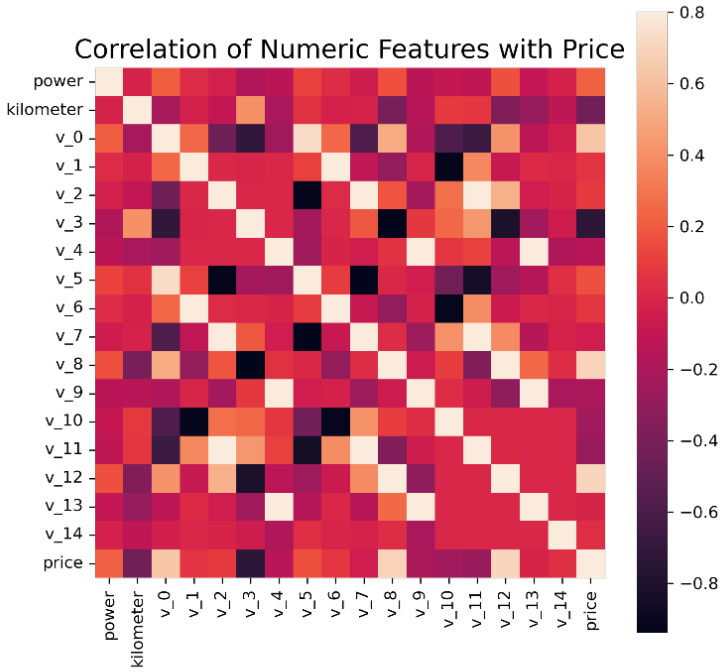

Figure 3 This heatmap shows the correlation between each numerical feature to the target variable (price). V_0, V_3, V_8, V_12 are highly related to the price. So the new features construction may depend on these 3 features. Those positive correlated features will be the important ones that need to pay extra attention in the analysis.

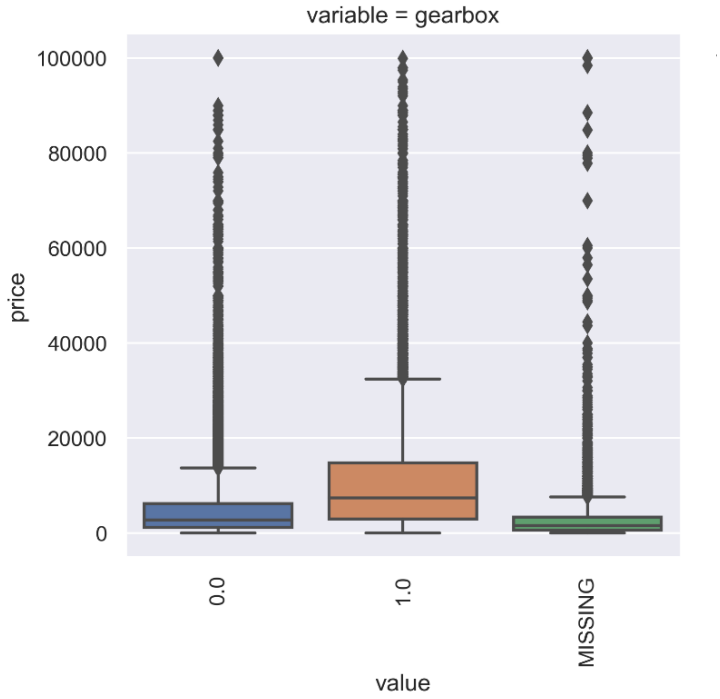

Figure 4 The box plot shows the 2 types of gearbox, 0.0 stands for manual shift, 1.0 stands for automatic shift. And the missing values also labels out. From the plot, there are a lot of outliers exist, but delete them directly is not a good idea since the impact of those value to the target variable is unknown. However, this will still give us an overview of how's the outlier distributed in the dataset for each feature.

3. Methods

3.1 Data Splitting and Preprocessing (Pipeline starts)

After the EDA steps, the feature engineering and data preprocessing should be introduced to the dataset now. The dataset is IID, which means each trading price are independent from next one. The price has a right-skew distribution, so perform log transformation on it to get it normalized. Concatenation of train and test dataset for future feature engineering. The following sentences shows some detail of feature engineering. First new feature will be the cars used time, generally, it is inversely proportional to the price. And there are 15k datapoints are missing, but since it's over 7.5% of the dataset, so we keep it. Also constructs feature city from regionCode, and each brand's sale amount and so on. then perform data bucketing to the training data. Delete the features that are used to construct new features. Furthermore, preprocess the brand and price related numerical features with Minmax scaler, and OneHot Encoder for the categorical features. Also, using mode(0) to fill the missing value. Run a reduced memory function for the new data, decreased 73.6% memory for faster processing in prediction. After preprocessing there are 139 features. Since the preprocessing is done above, a self-defined MLpipe_KFold_R2 function in splitting data step includes: for

each random state (5 in total), split data into 20% of testing and 80% of training and 5-KFold is applied to training data as the cross validation.

3.2 Pipeline Continues

To continue after the train, test set separation in above steps, the algorithms are introduced in the pipeline now (the algorithms are also performed under different random state). Add algorithm to the make_pipeline function and prepare the grid search for the finding the best parameters of each model, cross-validation will be using KFold defined above. Since the dataset has many independent variables, R2 score is chose to be the evaluation metric, to show the goodness of fit of the models.

3.3 Model Comparison

Based on the problem type (regression), 6 regression models are chosen and trained: Linear Regression with L1 regularization, Linear Regression with L2 regularization, K Neighbors regression, Random Forest regression, LGBM regression, and XGBoost Regression. Random Forest parameters are the default one (not tuned) since the dataset size is too large (running it will cause kernel died). The rest models' parameters are tuned by GridSearch with cross-validation method to find the best parameters. The table below lists the parameters in the tunning process.

Model Name	Parameters tuned
Linear Regression with L1 regularization	**Alpha**: np.logspace(-5,5,25)
Linear Regression with L2 regularization	**Alpha**: np.logspace(-5,5,25)
K-Neighbor Regressor	**n_neighbors**:1,11,30,100
LGBM(Light Gradient Boosted Machine) With gbdt (gradient Boosting Decision Tree)	**max_depth**: -1,1,2
XGBoost	**max_depth**: 2,3,4,5,8 **subsample**: 0.75, 0.8

Table 1. The parameters for each model that the code tried are listed above. The LGBM's max_depth include -1 since decrease max_depth is a way to increase the LGBM model performance.

3.4 Consideration of Pipeline Steps

Even the different random state applies to the models, the standard deviations of the scores are very small (below 0.4). For the random forest, the randomness is due to the selection of random subsets of the features to learn on, this is another reason not to choose RF. Rest steps of the pipeline building looks reasonable and good due to the experience.

4. Results

4.1 Model Selection

The tunning process returns the best parameters for each model. The average R2 score can be then compare between each model. The follow figure shows the average R2 score for each best model that generate from each random state.

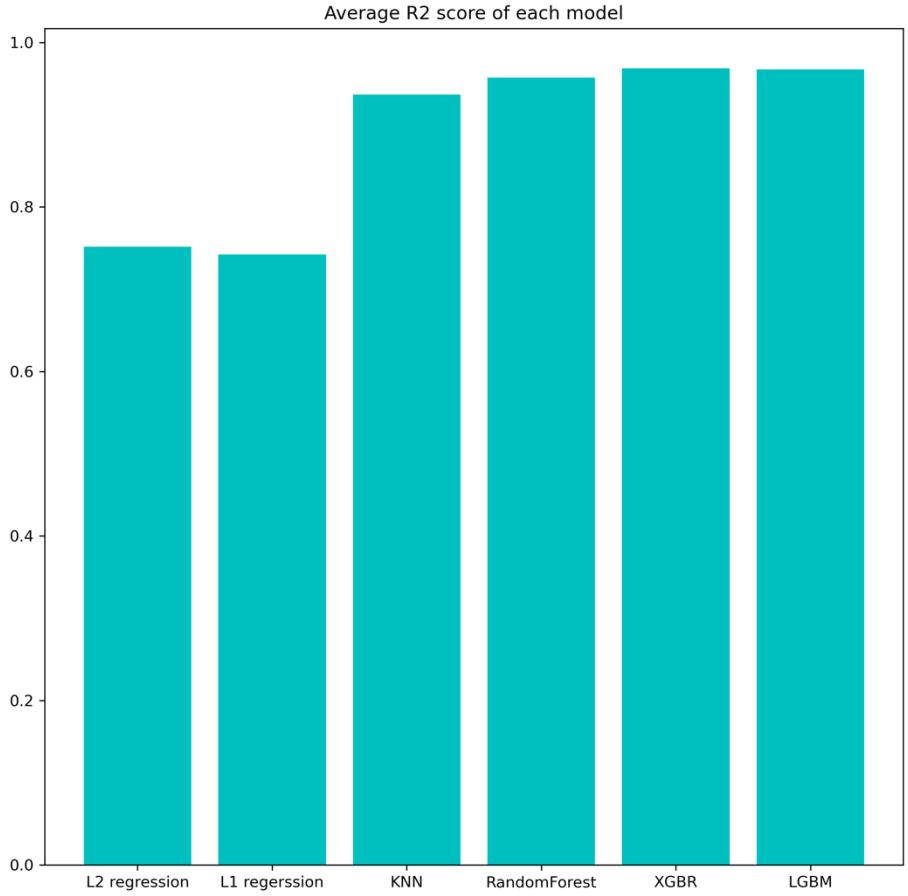

Figure 5. The above histogram shows the average R2 score of 6 selected models. The XGBR has the highest score 0.9686 which is slightly higher than LGBM's score 0.9672.

After reading the plot, the best model choice should be clear. However, if based on the length of running time, the 2 linear regression models are the best. But the final choice consideration still based on the R2 score, so the XGBR is most predictive model.

4.2 Score Comparison and Feature Importance

Baseline average R2 score 0.5449 with standard deviation 0.014. The trained XGBR has an average R2 of 0.9687 with standard deviation 0.0009, so the trained model R2 score is 30 std above the baseline score.

The global feature importance is calculated for XGBR, LGBM and RF. Below is the XGBR global feature importance plot.

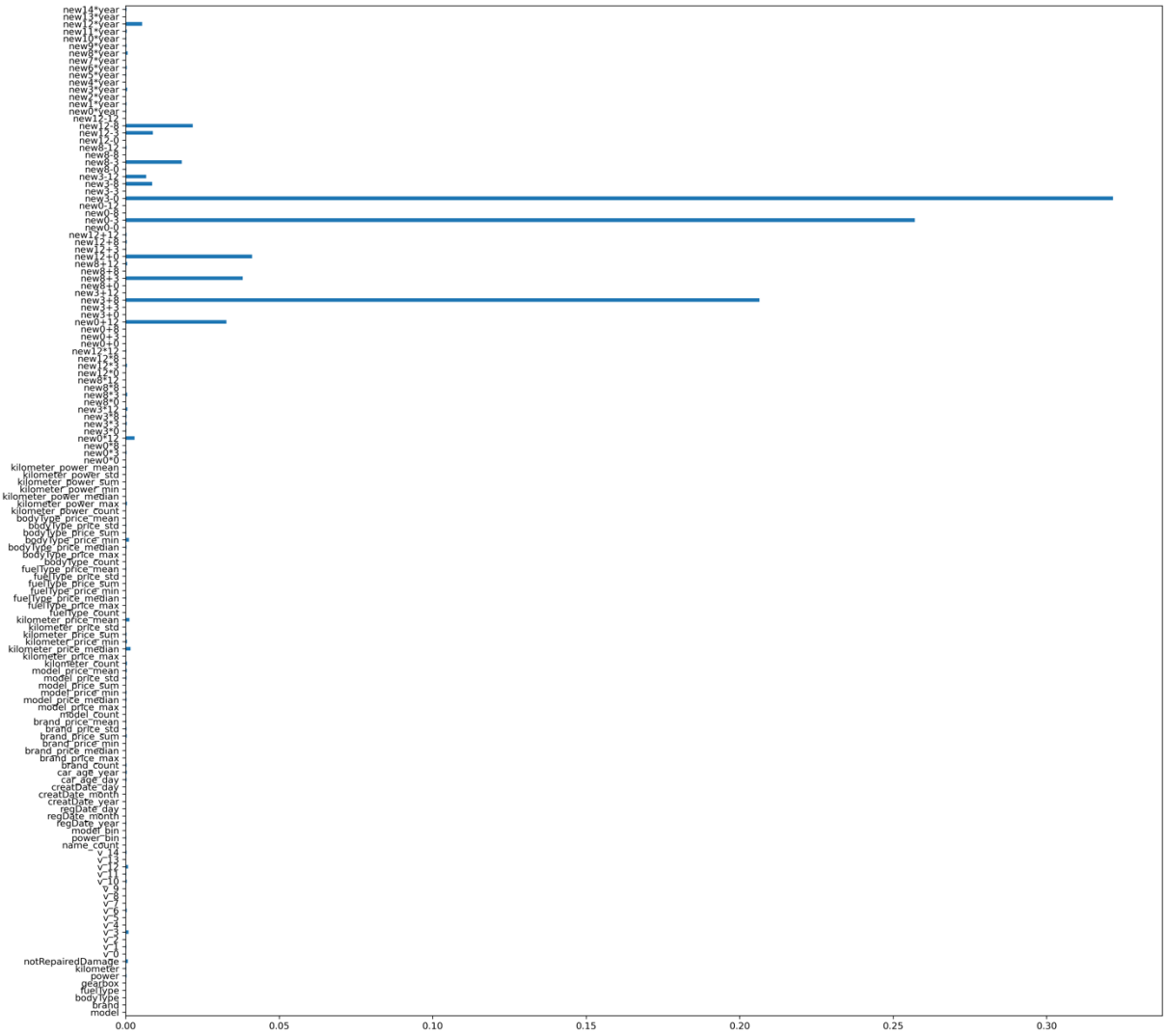

Figure 6. According to the plot the top 3 important features in XGBR are the new constructed: V3+V0, V0+V3, V3+V8. There are 10 features have zero feature importance. The top 10 important features are generated by the feature engineering.

Weight	Feature
0.2692 ± 0.0013	new3-0
0.2471 ± 0.0012	new0-3
0.0213 ± 0.0007	new12*year
0.0173 ± 0.0003	new8*year
0.0130 ± 0.0001	new8+3
0.0120 ± 0.0002	notRepairedDamage
0.0108 ± 0.0003	kilometer
0.0102 ± 0.0001	new3+8
0.0079 ± 0.0001	v_14
0.0060 ± 0.0001	new11*year

Figure 7. This is RF regression's global feature importance. The top three features are built in feature engineering, which is V3+V0, V0+V3, V12+year.

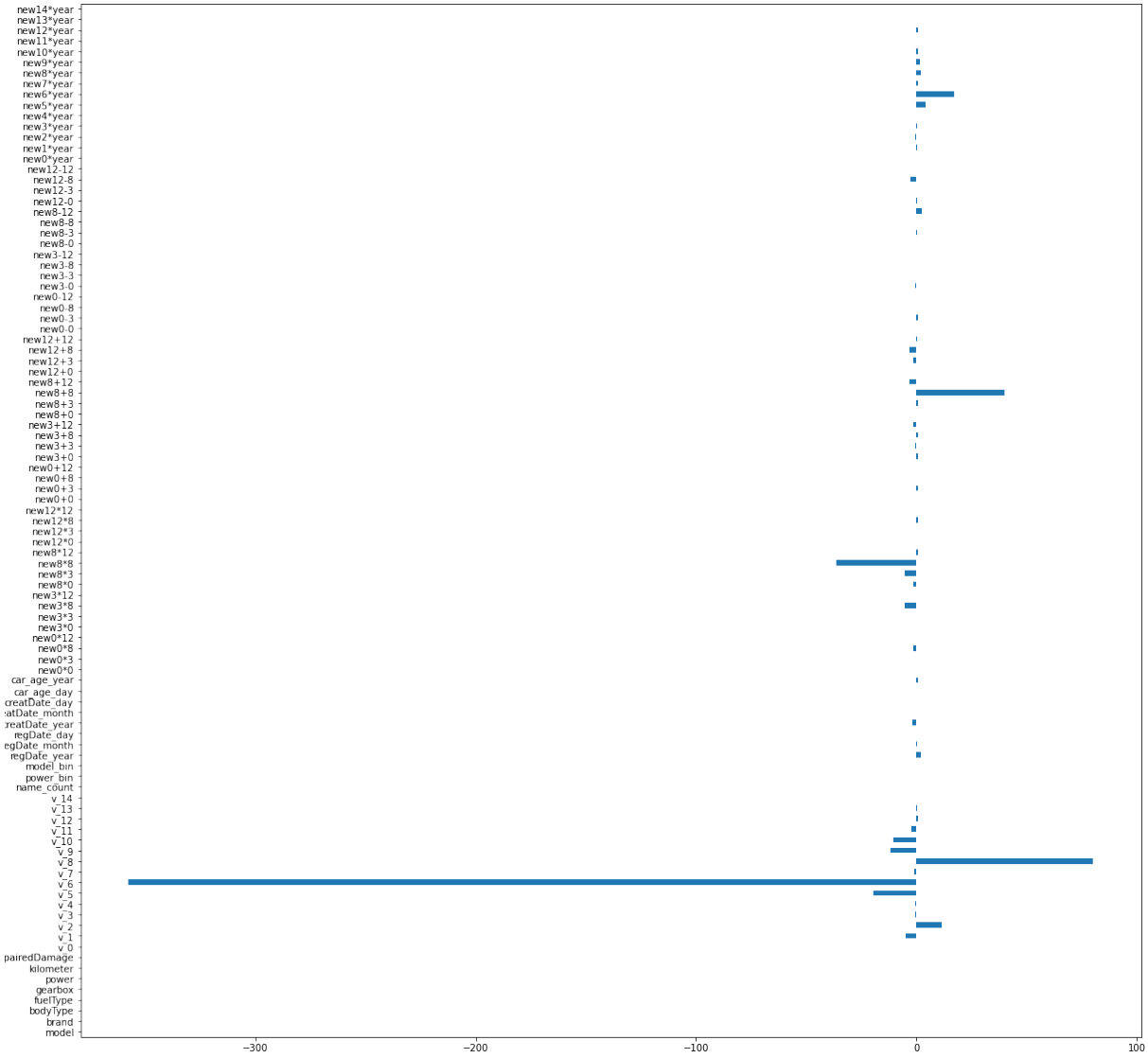

Figure 8. The global feature importance for LR(L1) is the coefficient value of each feature, the top 3 important features are V8, V8+V8, V6+year.

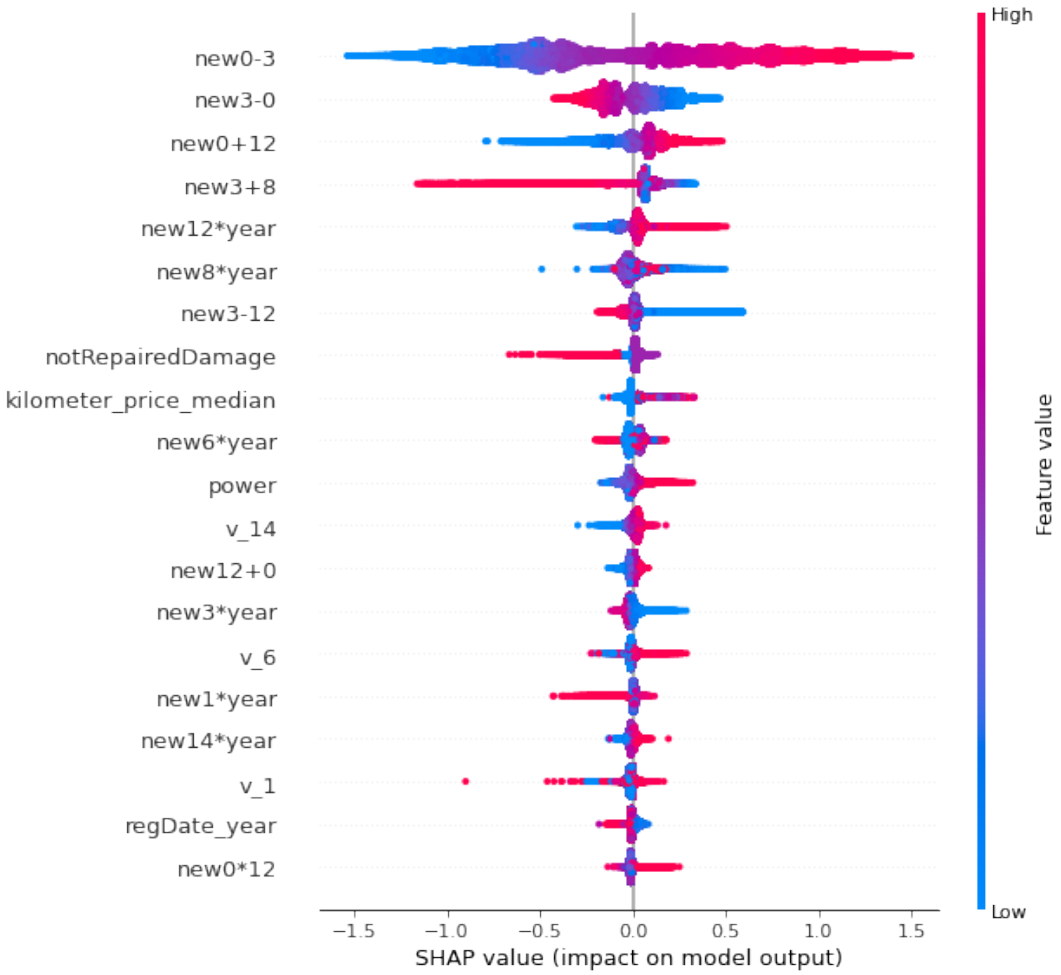

Figure 9. The SHAP value for 20 features of XGBR model global features is shown above. Basically, all features listed above has both negative and positive impact on the model prediction, like V0+V3(new0-3) largely impacts the model output both positively and negatively.

4.3 Model Results Interpretation

After going through the 3 models' feature importance graph, V3+V0 has a large weights in global feature importance. This makes sense since both component features ranks high of the correlation to price. One thing happens unexpected is that the LR models compute the 2 lowest R2 score, the problem looks like a simple linear regression like the famous Boston housing price problem. The Different from RF and XGBR, the V12+year are in the top 3 in LR(L1) model. There are several original features are in the top rank in LR(L1) feature importance, this maybe why the R2 score of LR(L1) is lower than other models. The model doesn't have as good performance as other models in feature and target variable correlation interpretation. The LR is too simple to interpret the dataset with many features. That validates that LR should not be chosen for this regression problem.

The SHAP value only computed for the first 20 features that impacted on model output most instead of local feature importance due to the data size the processing time. Even, the SHAP values for local feature importance are not able to be processed, the plot above gives us

lots of addition information of how those 20 features impact model outcome, which is not shown in the global feature importance graph.

Recall the original goal, the XGBR is applied to the original test set with 50k data size. And the R2 score is 0.986. It means, the model fits 98.6% of the data. The model works well with the parameters tuned and the price that it predicted has a high accuracy compared to the actual price.

5. Outlook

Given the results of LGBM, the max_depth can be tuned to more negative integer to improve the model performance. Then the final model will be LGBM since it can run 7 times faster than XGBR with a higher score. The XGBR can be run on GPU which will improve the speed and save time for more parameters' tunning process. In addition, the XGBR also can add alpha to it to make it runs faster under this high dimension situation, and more large numbers can be introduced to the max_depth to make the model learns more specifically to develop a more complex model. The SVM regression also can be implemented for this problem since SVM fits well for the big data set. Next time, the MAE score maybe used for model evaluation, smaller MAE it has, more accurate it is.

6. Reference

1. "天池_二手车交易价格预测数据分析." *开发者的网上家园*, www.cnblogs.com/cgmcoding/p/13279789.html.
2. *零基础入门数据挖掘 - 二手车交易价格预测赛题与数据-天池大赛-阿里云天池*. tianchi.aliyun.com/competition/entrance/231784/information.
3. Andrew Lukyanenko. (n.d.). *Predicting molecular properties*. Kaggle. Retrieved December 7, 2021, from https://www.kaggle.com/c/champs-scalar-coupling/discussion/96655.