MATH179 Midterm

Hoang Chu

1 Introduction

As a bike lover, I have no idea which car I should buy after graduation. Many people advised me to buy a car that has a good Return on Investment i.e resell-able in the future with the best price. But what resale price should I make for my used car? What elements should I look for to maximize the price? Will it be the brand of the car, or the historic usage of the car, or some specific elements of the car, or the listing location, or even the ZIP code of the dealer, or a combination of some of them? Fortunately, there is a dataset on Kaggle for values of used cars in the US (https://www.kaggle.com/datasets/ ananaymital/us-used-cars-dataset/data) with 3 million observations and 66 features, including the 'price' feature which will be our response values. Unfortunately, from the metadata, only 32 / 66 features having a description, hence I cannot rely on cherry-picking the feature based on domain knowledge.

2 Evaluation Metric

My metric will be the mean of (absolute error * 1.2 if under-pricing, absolute error * 0.8 if over-pricing) instead of any squared errors, since I deem that given the same absolute error, the consequence of under-pricing the value is not equal to over-pricing the value (as you can post your car again with a lower price).

3 Exploratory Data Analysis

Because of the large file, I read data in chunk sizes of 100,000 and merged them back, and also used Python to get the cleaned dataset with only compute-able data types, no Null values, and no duplicating rows / columns.

Read data

```
dfs = []
for number in range(30):
    start_row = number * 10**5
    end_row = (number + 1) * 10 ** 5
    df = pd.read_csv("~/Downloads/used_cars_data.csv",
                     skiprows=range(1, start_row + 1), nrows=(end_row - start_row), low_memory = False)
    dfs.append(df)
df = pd.concat(dfs, axis=ROW)
```

✓ 7m 2.1s Python

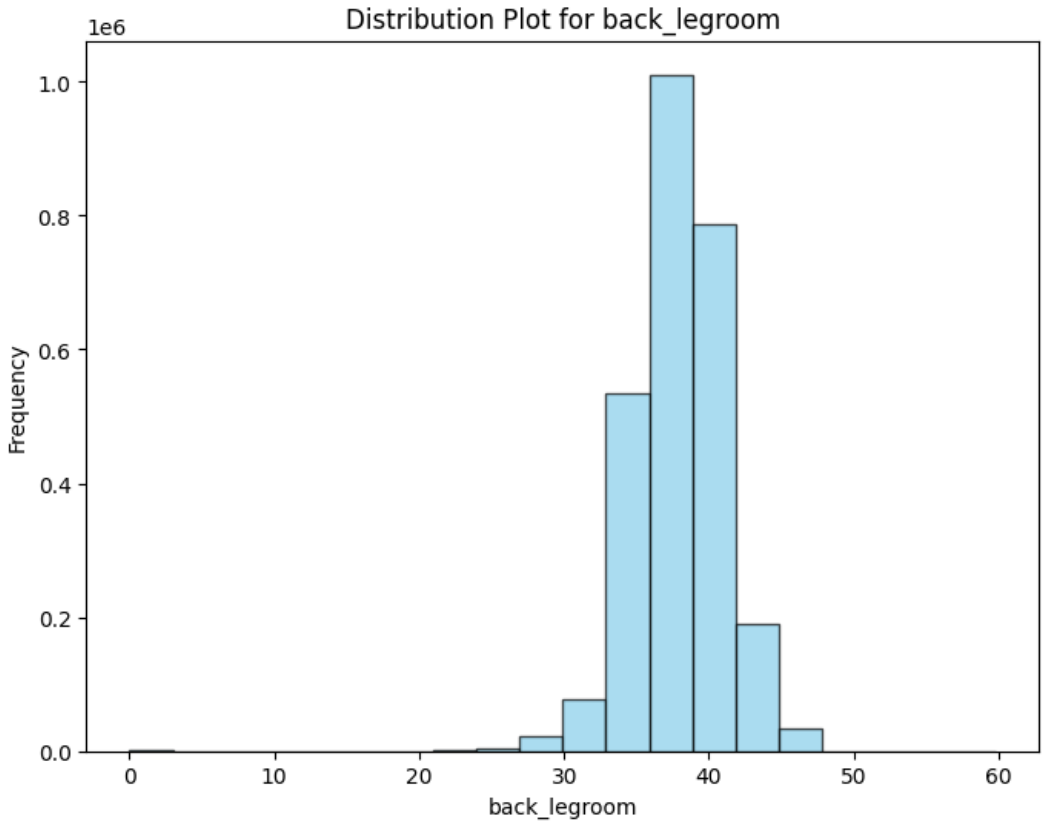
```
df.shape
```

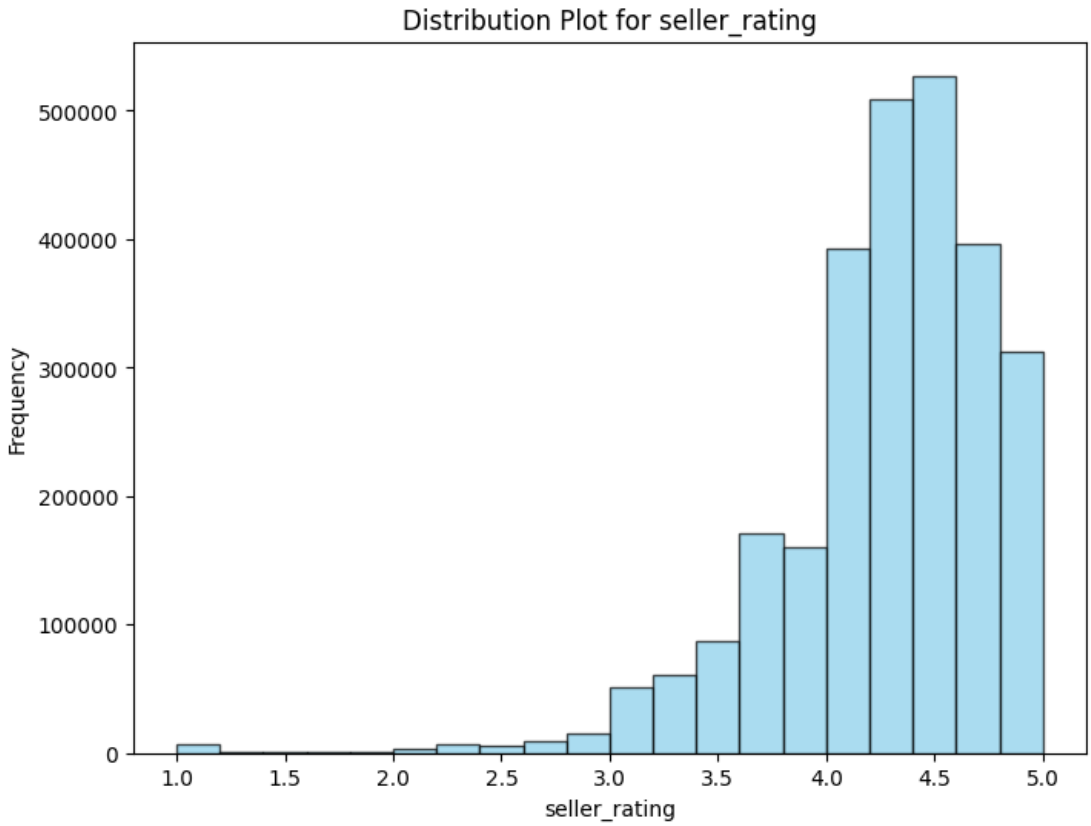
✓ 0.0s Python

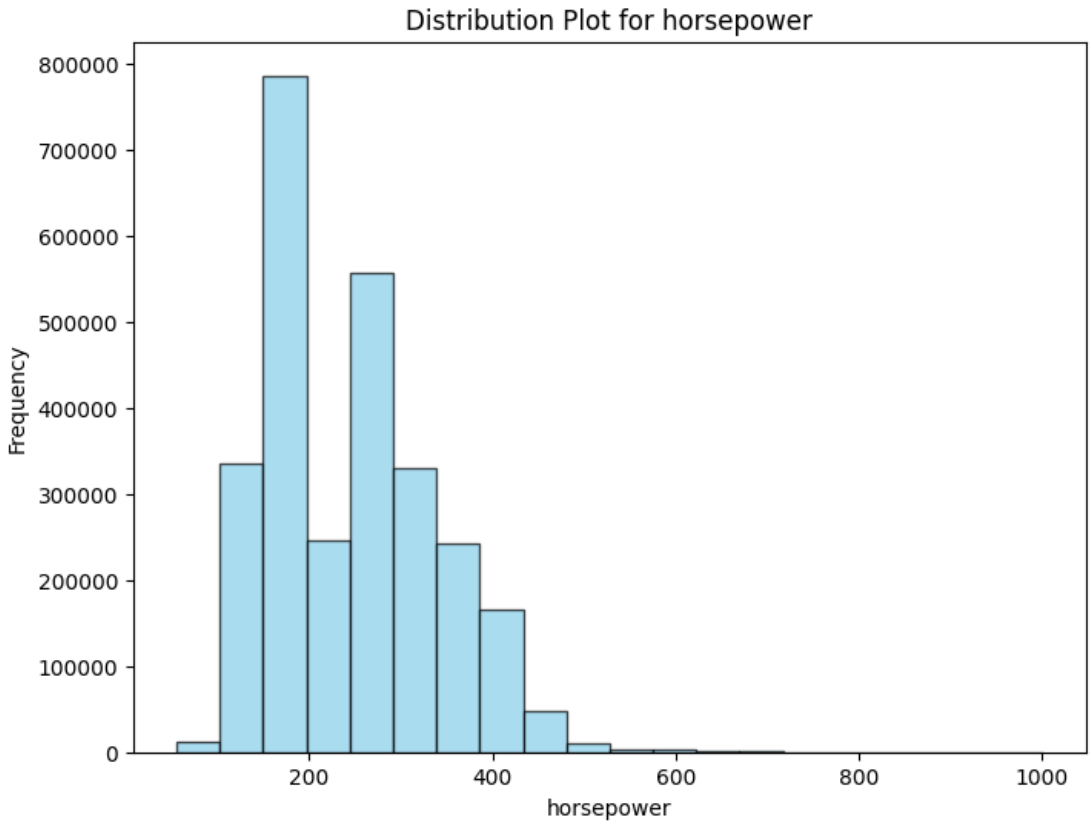
```
(3000000, 66)
```

3.1 Imputation

There are some columns having $< 20\%$ of null values left. Hence I decided to impute them by the column's maximum likelihood value, which is the mean for normal distribution, the median for skewed distribution, and the mode for other distributions (e.g bimodal distribution), as demonstrated below.




Distribution Plot for seller_rating
Frequency
0
100000
200000
300000
400000
500000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
seller_rating


Distribution Plot for horsepower
Frequency
0
100000
200000
300000
400000
500000
600000
700000
800000
200
400
600
800
1000
horsepower

3.2 Encoding

For columns having number of unique columns < 5, I will use one-hot encoding with new columns created ones since they will be col-linear with the intercept column, having full of 1. For columns having number of unique columns > 5, I will use frequency encoding and the labels will be based on the IQR of the frequencies. This means, $1 : 0 - 25\%, 2 : 25\% - 50\%, 3 : 50\% - 75\%, 4 : 75 - 100\%$.

4 Full Linear Model

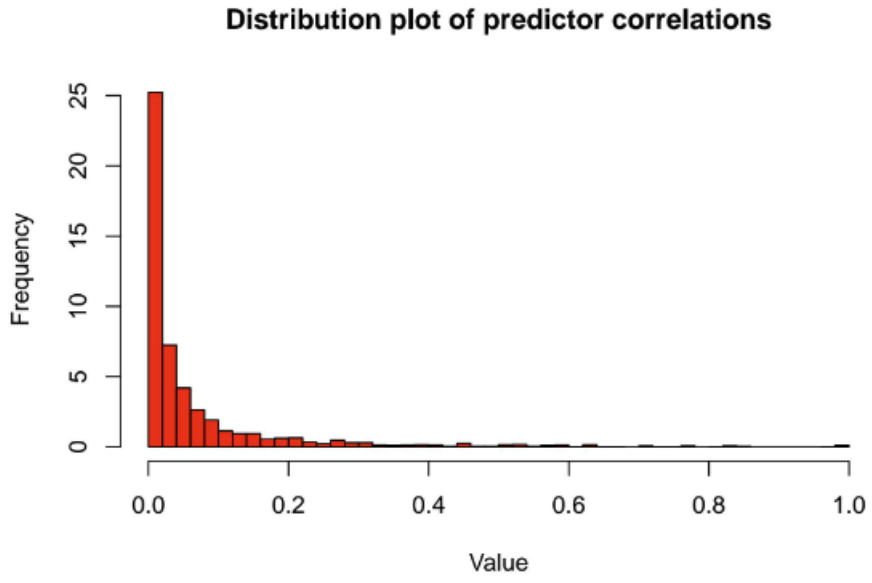
When running a full linear regression model, I got $R^2 = 0.67$, which is pretty good. However, the standard error of beta coefficient has a mean more than 298, indicating that the model is unstable. In other words, the beta coefficients are not uniquely defined, meaning the movement of response variable is not explain-able by the predictors via beta coefficients. This indicates extreme multi-collinearity between predictors.

5 Dealing with Multicollinearity

In class, we learned about Principal Component Analysis and Regularization as methods to solve multi-collinearity and reduce variances of the regression coefficients. Each of which has strengths and weaknesses, and I will explore them one by one.

For PCA, and regularization techniques, it's strongly advised to standardize both X and y as the penalty rate should be separated within columns, not between columns, and a penalty for being 1-level-too-high variance in column A should be the same as 1-level-too-high variance in column B. But I don't want to standardized encoded columns as they already been standardized by the encoding process earlier, and they're supposed to have integer data type.

Unfortunately, the majority of the predictor correlations are ¡ 0.2, meaning we will have a very sparse model using PCA (or even can't find a model at all) that's likely not have a good predictive power.



This means, the only method left is Regularization.

5.1 Regularization

Our only option left is Regularization to introduce bias to the model and reduce the variance from Bias-Variance trade-off, specifically Ridge regularization. However, as Ridge regularization never push regression coefficients to 0, if the regression coefficients are roughly close to each other, we can't know which predictors belong to what multicollinearity groups. An alternative is LASSO regression, which restricts the squares of the regression coefficients instead of the absolute value of regression coefficients and can push a regression coefficient to 0. Hence, a zero-value regression coefficients predictor indirectly indicates that it has lots of collinearity and should be removed.

5.2 LASSO-Elastic Net-AIC

However, as outliers of the dataset might still exist and influential and from the frequency distribution plot of coefficients, a lot of coefficients are concentrated in a range. Hence LASSO can still make our model to sparse just as PCA and PLS if the penalty term is too high. To mitigate this, I will divide the penalty term and combine a LASSO and Ridge regression aka Elastic-Net to reduce the variance of regression coefficients slower than purely LASSO, but still having the feature selection property of LASSO.

5.3 Optimized VIF Tracing

In addition to the tradition formula of VIF using R'2, the VIF of a predictor i is also the i'th diagonal value of the inverse of predictors' covariance matrix. As the predictors' covariance matrix of regularized regression is just OLS's + (penalty) * (identity matrix), we can trace the VIF score much faster.

As the VIF scores of regularized regression are deflated compared to those of OLS, I restricted the regularized VIF to a smaller bound. For the upper bound, I found the smallest bound having possible penalty term.

Finally, I got maximum VIF = 3.9. As regularization methods introduce bias to the objective error function, given the same set of regression coefficients, the error of regularized regressions will always be higher than that of Ordinary Least Squares. Hence, I won't use this model for prediction, but I will keep the predictors it filtered and used OLS for prediction.

6 Conclusion

1. The final predictors being used are:

```
"longitude"                       "make_name"
"mileage"                         "model_name"
"savings_amount"                  "seller_rating"
"width"                           "year"
"is_body_type_Sedan"              "is_body_type_Pickup_Truck"
"is_body_type_Convertible"        "is_fuel_type_Diesel"
"is_fuel_type_Flex_Fuel_Vehicle" "is_fuel_type_Hybrid"
"is_is_new_False"                 "is_franchise_dealer_False"
"is_wheel_system_FWD"             "is_wheel_system_4X2"
```

2. Given these predictors, I got mean training error = 4125.352 and mean testing error = 4988.026

3. There are a couple of improvements I hope to make. I think a big part I hope to improve is having a better encoding system as NLP can vectorize the feature importance of text relatively well. In addition, I would love a more rigorous analysis of dealing with outliers as there are various M-Estimation methods other than Huber Regression. I also wonder what other techniques can solve for Homoskedasticity violation.

That's all for my analysis! Thank you so much for reading.