PRICE PREDICTION OF USED CARS

ORIE 4741 PROJECT REPORT

Jiahe Chen [*]
jc3472@cornell.edu

Wei Cheng [†]
wc655@cornell.edu

December 6, 2021

1 Problem Definition

The question we want to explore from this dataset is how to predict the price of used cars based on features like make, year, wheel drive, etc. Although such kind of price prediction can be found everywhere on used car websites, these websites never release their prediction methods and we do not know which factors have the most impact on the price.

Answering this question can provide a more transparent guideline to customers and also help the car owners know which parts of the car should be maintained properly or when the car should be sold so that the owner can argue for a better price.

Answering this question can also help companies like local car dealer make better decisions on buying/selling used cars since car price can be very location specific. For example, cars with all-wheel drive (AWD) can be very popular in places with snowy winter due to the better performance in snow or slick surfaces.

2 Data Exploration

The dataset we use comes from Kaggle. The data comes from posts on the website between 2021-04-4 to 2021-05-04 and thereby can well represent the recent market of used cars.

The original dataset contains 426,880 rows and 26 columns. Among these columns, there are 25 features and 1 prediction target which is the selling price. 7 features including: posting_date, id, url, region_url, image_url, VIN, and county are taken out since they do not have significant contribution to the prediction. After dropping these features, there are 3 real-valued features and 15 categorical features.

Fig. 1a plots the population of missing values for top 10 features that have most missing values. We can see that size, cylinders and condition are the top 3 features in terms of missing data.

2.1 Data Description

Detailed descriptions of all columns are listed below. To help us better understand the data, we also visualize some critical features in Fig. 1.

1. **price**: Type: real-valued data. Fig. 1b shows the distribution. We can see that there are some extreme outliers in the dataset. Data points with extremely high/low prices are dropped since these prices barely reflect true prices.
2. **region and state**: Type: categorical text data. These two features are the city name and the state name. All cars are from U.S. After combining these two features, we found that these cars come from 426 different cities in 50 states plus the capital city.
3. **manufacturer**: Type: categorical text data. There are 42 different manufacturers. Fig. 1f plots the distribution. We can see that top 3 most popular manufacturers are Ford, Chevrolet, Toyota.

[*]School of Electrical and Computer Engineering, Cornell University
[†]Ann S. Bowers College of Computing and Information Science, Cornell University

4. **model**: Type: categorical text data. Indicates the model of the car, such as highlander, elantra, etc. There are 29667 different models. However, not all of them are valid. Some models are mistaken with types. Some models may look different but they actually point to the same one since there is no standard way for sellers to write down the model.
5. **condition**: Type: categorical text data. Fig. 1l shows that most cars are in 'good' or excellent' condition.
6. **cylinders**: Type: categorical text data. Car engine with more cylinders can provide more power but also consume more gas. Most cars have 4 to 8 cylinders as seen in Fig. 1g.
7. **fuel**: Type: categorical text data. There are 5 types: 'gas', 'other', 'diesel', 'hybrid', and 'electric'. Most cars run gas.
8. **drive**: Type: categorical text data. There are 3 types: 'rwd', '4wd', and 'fwd'. Fig. 1h shows the distribution.
9. **lat & long**: Type: real-valued data. The latitude and longitude of the location of the car. Fig. 1d shows that most cars are from California and Northeastern regions. Some data points have invalid coordinates and are dropped. Valid ranges of latitude and longitude are: $18°$ to $72°$ and $-177°$ to $-67°$.
10. **odometer**: Type: real-valued data. Indicates how many miles traveled by the car. Fig. 1c shows the distribution.
11. **title_status**: Type: categorical text data. There are 6 types: 'clean', 'rebuilt', 'lien', 'salvage', 'missing', and 'parts only'. Most cars have clean title.
12. **transmission**: Type: categorical text data. There are 3 types: 'other', 'automatic', and 'manual'. Most cars have automatic transmission as seen in Fig. 1j.
13. **size**: Type: categorical text data. There are 4 types: 'full-size', 'compact', 'mid-size', and 'sub-compact'. Most cars have full size.
14. **type**: Type: categorical text data. Fig. 1k shows the distribution. Top 3 most popular types are: mini-van, pickup and SUV.
15. **year**: Type: categorical data. The year can range from 1900 to 2022. Fig. 1e shows that most cars have year in 2000 to 2020.
16. **paint_color**: Type: categorical text data. Fig. 1i shows that top 3 most popular colors are white, red and black.
17. **description**: Type: text data. The text description of the car.

3 Data Preprocessing

3.1 Missing Values

Since we want to evaluate the impact of all features and we have sufficient amount of data, we dropped all data with missing values. This leaves us around 79,016 data points to be used for prediction.

3.2 Invalid Values

Since our data are crawled from Craigslist, the data is very messy and contains some invalid values. We will discuss how we process those invalid values for each feature in the following:

1. **price**
   There are some extreme outliers in the price column, because of people randomly putting numbers, such as 0, 1, or 123456789. Therefore, we decided to only look at price from 0.25% to 97.5%, and drop the 5% data points with extreme price values.
2. **model**
   Some people misunderstood the meaning of car model, and put invalid values, such as "SUV", "truck". Therefore, we removed data points whose model appears less than 10 times in the dataset.
3. **latitude and longitude**
   Some latitudes and longitudes indicate locations outside US, which is out of the scope of our project. So, we removed those data points.

3.3 Feature Encoding

We used one-hot encoding for all categorical features throughout our analysis.

3.4 Normalization

We used Z-score Normalization throughout our analysis. So, each feature column has zero mean and unit variance.

4 Machine Learning Methods

4.1 Car manufacture-based Model

Since our data has a very large number of data points($\sim 420k$), it will take too long time for machine learning models to converge. Even worse, there are too many categorical features in our data, and if we one-hot encode all the categorical features, we will have over 30,000 features, which will cause over-fitting. Therefore, we decided to train individual model for each car manufacture.

4.2 Toyota Example

There are 42 car manufactures in our dataset, for simplicity, We chose Toyota as our example, and separate out all the Toyota data points to make a new dataset. For the rest of this report, our analysis will all be on the Toyota dataset, but this method can be easily generalized to all car manufactures.

The Toyota dataset contains 13,095 data points. We trained 6 machine learning models on it: linear regression, ridge regression, lasso regression, huber regression, random forest, and boosting. For this method, we used the following 11 features to achieve the best result: odometer, region, year, model, cylinders, condition, fuel, title_status, transmission, drive, paint_color.

4.3 Machine Learning Models

Details about each model we trained are given below:

1. **Linear Regression**[1]
   Linear regression uses L2 loss function and no regularization. Therefore, it over-fitted as we could expect. Our linear regression achieved training RMSE = 3220, and testing RMSE = 3633.
2. **Ridge Regression** [2]
   Ridge regression uses L2 loss function and L2 regularization. Therefore, it has a better testing RMSE as compared to linear regression. It has a training RMSE = 3220, and testing RMSE = 3520.
3. **Lasso Regression**[3]
   Lasso regression is similar to ridge, but uses L1 regularization instead. It has a training RMSE = 3232, and testing RMSE = 3504.
4. **Huber Regression**[4]
   Huber regression uses huber loss function and L2 regularization. It is robust to outliers. It has a training RMSE = 3318, and testing RMSE = 3430.
5. **Random Forest**[5]
   Random Forest is a tree-based algorithm. It uses bagging to combine multiply decision trees. Our random forest model has a better testing RMSE than the previous linear models. It has a training RMSE = 1199, and testing RMSE = 3152.
6. **Boosting**[6]
   We used decision trees as the base model for boosting. It has a similar testing RMSE to random forest. It has a training RMSE = 2606, and testing RMSE = 3126.

The table below summarizes the models, and their training and testing RMSE.

Algorithms	Training RMSE	Testing RMSE
Linear	3220	3633
Ridge	3220	3520
Lasso	3232	3504
Huber	3318	3430
Random Forest	1199	3152
Boosting	2606	3126

The best testing RMSE is achieved by the boosting algorithm, which has a testing RMSE = 3126.

4.4 Model Bias & Variance

The table below lists the variance and squared bias for each models. We did not list linear regression here, since it does not have regularization, its variance and squared bias are significantly larger than other algorithms.

Algorithms	Squared Bias	Variance
Ridge	13861	8944
Lasso	13093	7953
Huber	13777	7242
Random Forest	14205	3412
Boosting	12978	2356

From the table above, we can see that all linear models have similar variance and squared bias. For bagging(random forest) and boosting algorithms, the variance is smaller than the linear models.

4.5 Linear Model Coefficients

The table below lists the features whose absolute value of coefficients is among the top 5 largest, for each linear model. Those features are strong indicators for the car price.

Algorithms	Top5 Features
Linear	odometer, 2018, 2017, tacoma double cab pickup, 2016
Ridge	2017, 2018, 2016, tacoma double cab pickup, 2019
Lasso	2017, 2016, 2018, 2019, tacoma double cab pickup
Huber	odometer, 2018, 2017, tacoma double cab pickup, 2016

From the table above, we can see the top 5 features for each model are very similar. Although one-hot encoding makes it difficult for us to interpret, we can see all top5 are from odometer, year, and car model, which totally makes sense when it comes to estimate an used car's price.

5 Car Modelwise Method

In our previous method, we fund that there are 1,722 car models in the dataset. One-hot encoding the car model created a lot of features, which may cause over-fitting. Also since car model is a very significant feature in our prediction, instead of just one-hot encode them, we also tried to train separate machine learning model for each car model. We call it "modelwise method".

Since the number of training data points for each model is fewer than before. In order to avoid over-fitting, We only included 7 features for this method: odometer, lat, long, year, paint_color, condition, title_status.

The table below shows our modelwise results, together with our previous results.

Algorithms	Training RMSE	Testing RMSE	Training RMSE(modelwise)	Testing RMSE(modelwise)
Linear	3220	3633	2069	59184
Ridge	3220	3520	2196	3175
Lasso	3232	3504	2392	3842
Huber	3318	3430	2350	3565
Random Forest	1199	3152	1180	3047
Boosting	2606	3126	1132	3067

From the table, we can see that compared with our previous method, in the modelwise method, each model over-fit more, due to fewer data points for training each model. However, the modelwise method generally has a better testing RMSE than before. Ridge, random forest, and boosting now all have a better testing RMSE. The The best testing RMSE is achieved by the random forest, which has a testing RMSE = 3067.

6 Fairness

By the nature of our project, we don't think fairness is applicable. There isn't any protected attribute in our dataset, and we don't think we can have any proxies either. Therefore, we think our project shouldn't have any fairness issues.

7 Conclusion

In conclusion, our model can achieve test RMSE around 3000. Considering that used car's price are typically in tens of thousands, and the price of used cars are varied in a typical range(no one can tell the exact value for an used car), we think our model performs very good. Due to the limited time, we did not have chances to make some further improvements to our model, like spending more time tuning model parameters. We think after those improvements, We are confident that our model can be used in production to help customers make the best decisions on buying/selling used cars.

References

[1] [Online]. Available: https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html

[2] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

[3] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

[4] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.HuberRegressor.html

[5] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[6] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

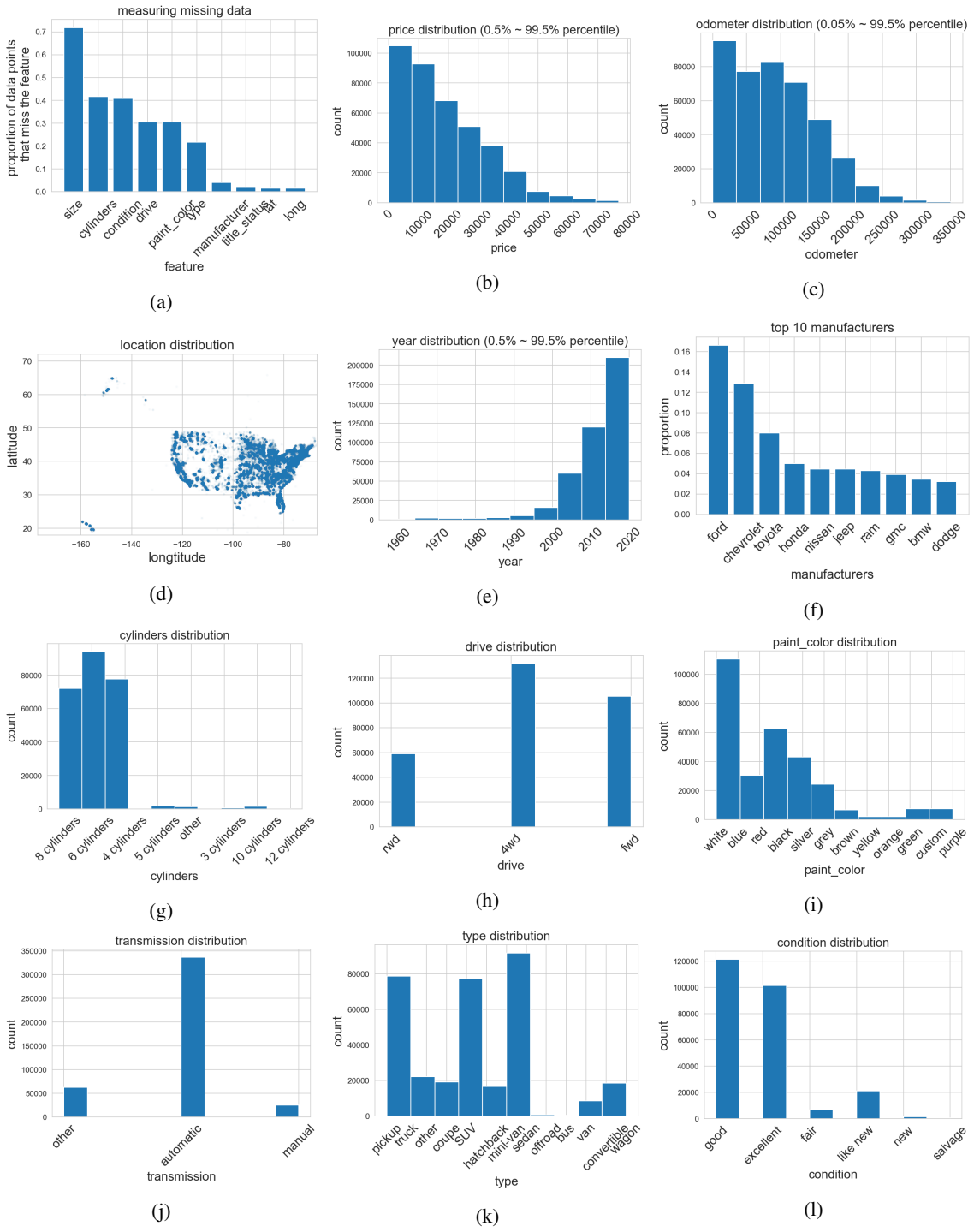

Figure 1: Visualizations of the data.