Used Car Pricing and Beyond: How Data Analytics Increases Transparency and Equality for Dealership

Mi Zhou, Liyang Du
mz558@cornell.edu, ld477@cornell.edu

Abstract

A significant part of the overall automotive market is derived from the used car trade. By utilizing the world's largest web listing: Craigslist as a proxy data source, we can create models for the used car pricing based on the asking prices listed in the web adverts. This type of data acquisition requires a thorough data cleaning process to generate dependable statistical models after all. We proposes several well-defined supervised learning approaches to study how they can help determine the market values of the used-cars from this type of big messy data. The models can generalize used cars' prices very well by choosing a subset of the 16 variables we take into consideration.

1 Introduction

There are a variety of features of a used car such as age, maker, origin (the original country of the manufacturer), mileage, horsepower, the type of fuel the car uses, style, braking system, cylinder volume, acceleration, number of doors, safety index, size, weight, height, paint color. The price is estimated based on the number of features as mentioned above. We are planning to deploy a decision support tool to aid buyers and sellers for assessing their cars market values. This deployed service can also aid to determine the prices of other used items through incorporating the features and data input by the users. In summary:

- We collect the data about used cars from Craigslist, identify important features that reflect the price.
- We preprocess, fill or remove entries with empty or NA values. Discard features that are not relevant for the price prediction.
- We apply several supervised learning algorithms on the preprocessed data with designated features as inputs and the estimated price as output.

The rest of this paper is organized as follows: Section 2 describes the dataset; Section 3 describes the approaches we tried to tackle the problem and our concrete algorithms; Section 4 lists the results we obtained from linear regression and discusses how confident we are in it. Section 5 lists the results we obtained from tree-based algorithms and discusses how confident we are in them; Section 6 demonstrates why we should use our models in production to help decision-making; Section 7 argues that our project would not produce a Weapon of Math Destruction since algorithmic biases are minimized; Section 8 presents suitable metrics to measure fairness in our application.

2 Dataset

2.1 Data Characteristics

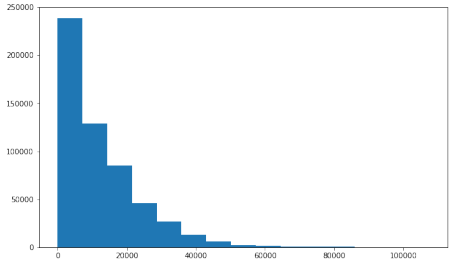
The dataset was scraped from Craigslist. It has 550313 rows and 22 features on used car sales,

containing most all relevant information on car sales including price, condition, manufacturer, latitude/longitude, and 16 other categories. The output that we wish to predict is the price of the used car. Out of the 22 features, there are 3 url type features (url, city url, image url), 5 numeric type featuers (price, year, odometer, latitude, longitude), 13 categorical type features (city, manufacturer, make, condition, cylinders, fuel, title status, transmission, VIN, drive, size, type, paint color), and 1 text features (desc - meaning description). We hope that such a large dataset will lend itself to the creation of a state-of-the-art pricing model.

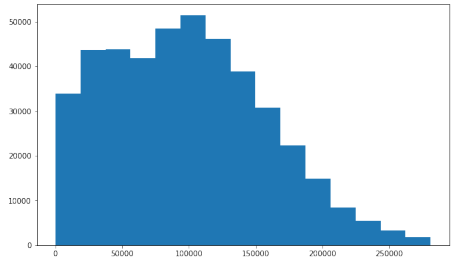
Out of the 22 features, there are 17 features having missing values except for city, price, url, city url, image url. The features that have missing ratio higher than 20% are condition, cylinders, odometer, VIN, drive, size, type, paint color. The count of missing values are: year: 1487, manufacturer: 26915, make: 9677, condition: 250074, cylinders: 218997, fuel: 4741, odometer: 110800, title status: 4024, transmission: 4055, VIN: 239238, drive: 165838, size: 366256, type: 159179, paint color: 180021, latitude: 11790, longitude: 11790.

2.2 Exploratory Data Analysis

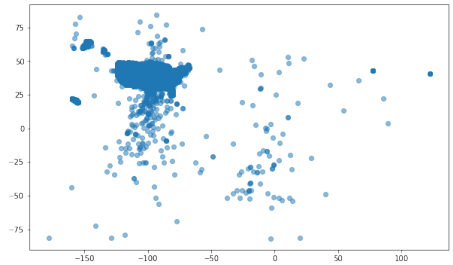
We visualize some features about our data. First we look at the price histogram. The price is extremely heavy-tailed: some cars have extremely high prices. So we plot the histogram of the price of cars with the lowest 99.9% price. Also note that there are around 600 cars that have extremely price above 99.9% of the cars.



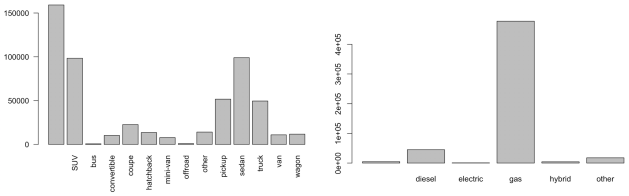
Then we look at the odometer, which we consider is an important feature in determining the price of the used cars. After dropping NA values, the odometer is also extremely heavy-tailed and we only plot the histogram of the lowest 99% odometers.



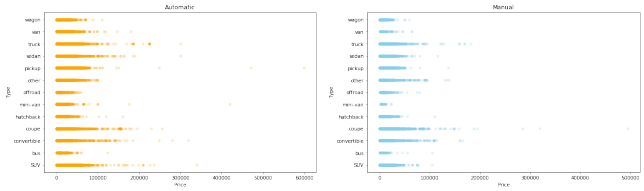
Another interesting thing in this dataset is that it provides the longitude and latitude of every trade. Since most price predictions do not consider using these features, let us have a look of the geographical property of the dataset.

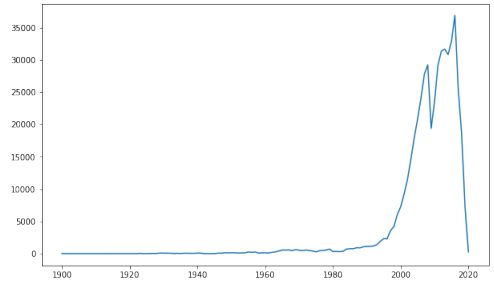


Then let us look at some categorical features. Since car types and fuels might be a good impact on the price, we look at the fequency plot of the types and fuels. Note that the most left column stands for NA values.



We might also look at some joint distribution of price and some features. This way we can have an intuition of how one feature is correlated with price. So we look at the joint distribution of the price and car types under 3 different transmissions (automatic, manual, other).





2.3 Data Preprocessing

In this part we illustrate our approach in data cleaning and preprocessing. The **first step** we adopt is dropping features. We drop the 3 url type features (url, city url, image url) because in our models the url features are hard to use. Also we drop the longitude and latitude features because we think there does exist the relationship between location and price, but this relationship is not linear in longitude and latitude which are numerical values, because there are many places of high price and low price but they are not linear in location. Also we drop the text feature (desc) because it is also hard to use in the model. Now we have 16 features: 1 price to be predicted by the other 15 features.

The **second step** is dealing with missing values. If one row has more than half features missing (i.e. more than 8 features) then we would drop this row. This is because if one data point has many missing features then it is rather inaccurate to incorporate it into the model.

The **third step** is dropping rows. We drop rows of 0 prices and prices over 100,000. We think 0 prices records are useless points in our price-predicting model so we drop 0 prices points. As we we in the exploratory data analysis part, there are some extreme price records (around 0.1% of the records of prices higher than $100,000) which might introduce high bias to the model. So we drop these rows. Then, if we look at the following year distribution, we will find that the records after year 1999 and before year 2020 have more than 7000 records each year. We believe the number of points within each year should not be too small to give a stable model, so we keep records after year 1999 and before year 2020.

Then we keep rows of odometer smaller than 500,000. This is also to avoid introducing bias because the records are few and the prices are unstable. Finally, we drop all rows with missing features if the missing ratio of these certain features are higher than 50%. Though this is a bold decision, we find that there are features as size, condition and cylinder that have high missing ratio. We do not want to impute such a huge missing ratio to introduce high bias. For features with small missing ratios, we impute the missing values by mean (numeric features) or most frequent values (categorical features).

The **last step** is replacing the city feature, where there are more than 400 different cities, by the states they belong to (shrunk to 50 values). This is because we believe the cities are important in predicting price but if the categorical values are too many then the model will be slow and unnecessarily complicated. So we make the model simpler and faster to fit. Then we are ready to fit different models.

3 Algorithms

3.1 Linear Regression

We first applied multiple linear regression to predict the used car price as we assumed price is linearly dependent on various explanatory features such as millage, car type, maker and etc. The objective function is defined as:

$$p_i = \sum_{j=0}^{d} x_{ij}\beta_j + e_i, \quad i = 0, 1, 2, ..., n \qquad (1)$$

$$E(e_i|\{x_n\}) = 0 \quad and \quad E(e_i^2|\{x_n\}) = \sigma^2 \qquad (2)$$

where p_i is the price whose expected value depends on the covariates x_i and e_i represents the error terms and is assumed to be independent between observations. σ is unknown and usually the covariate x_0 is a constant 1 and β_0 is the intercept. In the multiple linear regression model above, the parameters β_i and the variance σ^2 are to be estimated.

The linear model (the lm() function in R) can be used to do linear regression. A linear model has the form response terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. QR factorization (a direct rather than iterative method) is employed to solve linear least squares problems in lm().

3.2 Regression Tree

Classification and regression tree, abbreviated as CART, is an algorithm that uses a bisection recursive segmentation technique to divide the current sample set into two sub-sample sets, so that each non-leaf node generated has two branches. The features of non-leaf nodes are True and False, the left branch is True, and the right branch is False.

The CART regression tree predicts regression continuous data, assuming that X and Y are input and output variables, respectively, and Y is a continuous variable. In the input space where the training dataset is located, each region is recursively divided into two sub-regions and the output value on each sub-region is determined to build a binary decision tree.

For a given data set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i is and $m-$dimensional vector, the goal is to find a function that minimizes MSE:

$$\min \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

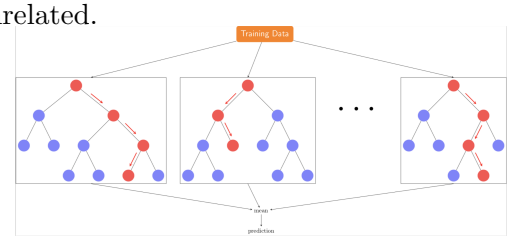
If a regression tree has M leaves, which means the tree partitions the input space into M units $R_1, ..., R_M$, then the tree has at most M predicted values, and the MSE becomes

$$\min \frac{1}{n} \sum_{m=1}^{M} \sum_{x_i \in R_m} (c_m - y_i)^2$$

For classic regression trees, the key idea is recursive partitioning. The process begins by taking all data for the consideration of all possible values of all variables for growing a tree. So it will select on variable or value that produces the best separation in the target attribute. If the value in focus is lower than the value at the separate point, that value will be placed on the left side of tree. For the value greater than or equal to the value at separate point, it will be sent to right side of tree. The tree will repeat this splitting process until it cannot find another best separate point the give the increase purity greater than the last separate point.

3.3 Random Forest

Random forest is an algorithm based on single regression tree, in that it uses resampled data to construct many trees and average them. For a given data set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, bagging method resamples from training set with replacement for B times, and trains tree models on these samples. After training, the averaged tree is used to conduct prediction. This bagging method decreases variance while keeps bias not increased. Random forest uses random $\sqrt{p}$ features if there are p features in total to keep trees unrelated.



3.4 XGBoost

XGBoost, short for Extreme Gradient Boosting, was introduced in [12]. Differing from the traditional boosting algorithms GBM that divides the optimization problem into two parts by first determining the direction of the step and then

optimizing the step length, XGBoost tries to determine the step directly by solving

$$\frac{\partial L(y, f_{m-1}(x) + f_m(x))}{\partial f_m(x)} = 0 \tag{3}$$

for each x in the training set. By doing second-order Taylor expansion of the loss function around the current estimate $f_{m-1}(x)$, we get

$$L(y, f_{m-1}(x) + f_m(x)) \approx$$
$$L(y, f_{m-1}(x)) + g_m(x) f_m(x) + \frac{1}{2} h_m(x) f_m(x)^2$$

where $g_m(x)$ is the gradient, same as the one in GBM, and $h_m(x)$ is the Hessian at the current estimate:

$$h_m(x) = \frac{\partial^2 L(Y, f(x))}{\partial f(x)^2} \tag{4}$$

Letting G_{jm} represents the sum of gradient in region j and H_{jm} equals to the sum of hessian in region j, the equation can be rewritten as

$$L(f_m) \propto \sum_{j=1}^{T_m} [G_{jm} w_{jm} + \frac{1}{2} H_{jm} w_{jm}^2] \tag{5}$$

With the fixed learned structure, for each region, it is straightforward to determine the optimal weight:

$$w_{jm} = -\frac{G_{jm}}{H_{jm}}, j = 1, ..., T_m \tag{6}$$

Plugging it back to the loss function, we get

$$L(f_m) \propto -\frac{1}{2} \sum_{j=1}^{T_m} \frac{G_{jm}^2}{H_{jm}} \tag{7}$$

Taking regularization into consideration, we can rewrite the loss function as

$$\begin{aligned} L(f_m) &\propto \sum_{j=1}^{T_m} [G_{jm} w_{jm} + \frac{1}{2} H_{jm} w_{jm}^2] \\ &+ \gamma T_m + \frac{1}{2} \lambda \sum_{j=1}^{T_m} w_{jm}^2 + \alpha \sum_{j=1}^{T_m} |w_{jm}| \\ &= \sum_{j=1}^{T_m} [G_{jm} w_{jm} + \frac{1}{2} (H_{jm} + \lambda) w_{jm}^2 \\ &+ \alpha |w_{jm}|] + \gamma T_m \end{aligned}$$

where γ is the penalization term on the number of terminal nodes, α and λ are for L_1 and L_2 regularization respectively. The optimal weight for each region j is calculated as:

$$w_{jm} = \begin{cases} -\frac{G_{jm}+\alpha}{H_{jm}+\lambda} & G_{jm} < -\alpha \\ -\frac{G_{jm}-\alpha}{H_{jm}+\lambda} & G_{jm} > \alpha \\ 0 & O.W. \end{cases}$$

Finally, the gain of each split is defined correspondingly:

$$Gain = \frac{1}{2} [\frac{T_\alpha(G_{jmL})^2}{H_{jmL} + \lambda} + \frac{T_\alpha(G_{jmR})^2}{H_{jmR} + \lambda} - \frac{T_\alpha(G_{jm})^2}{H_{jm} + \lambda}] - \gamma$$

$$T_\alpha(G) = \begin{cases} G + \alpha & G < -\alpha \\ G - \alpha & G > \alpha \\ 0 & O.W. \end{cases}$$

Instead of assigning different weights to the classifiers after every iteration, XGBoost fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. Thus, the model is being updated using gradient descent and hence gradient boosting. XGBoost specifically implements this for decision tree boosting with an additional custom regularization term in the objective function.

4 Linear Regression: Results and Confidence Analysis

4.1 Original Linear Regression

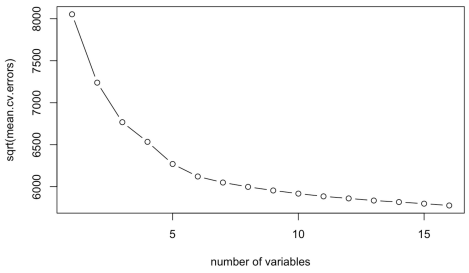
First we try the simplest linear regression, which means we do not include any variable selection, dimension reduction or parameter regularization in regression. We randomly partition the data into training part and testing part, which are 80% and 20% of the total data respectively, and then do the linear regression of all features on the training part.

The regression results show that the majority of the 16 features are very statistically significant (with p-value at least smaller than 0.01), whereas only some values of some categorical variables are not as statistically significant. We also get the adjusted R^2 0.6844 with root mean squared error 5486.919 on test set, which reveals that the model captures 68.44% linear relationship and the squared mean predicting error is around 5486.919. It seems that the model gives a satisfactory prediction. We then seek to improve the model.

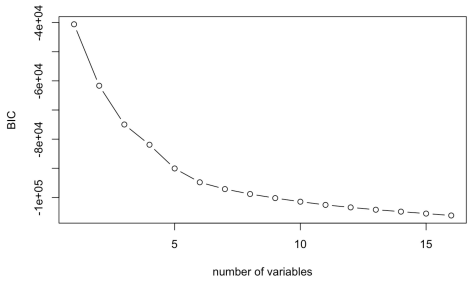
4.2 Remedying Overfitting: Variable Selection

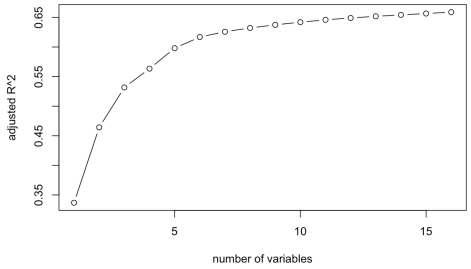
What would make the model imprecise when making predictions? From the testing error decomposition, we can decompose the testing error into bias, variance and noise. As noise exists all the time, we seek to deal with bias and variance. First, we consider is this model is too unbiased? We know a model too unbiased results from overfitting, which often happens when including too many irrelevant features or much higher order terms of features. Once overfitting is detected, we can address it by adding more data, using simpler models (cutting off more features and orders of features), adding regularizers or even using Bagging in decision tree.

As the p-value showed, the model has some irrelevant features in it. So we consider using some formal subset selection methods to see whether this model is highly biased (needing variable selection) more formally. We first apply forward subset selection with 5-folds cross-validation. The CV errors are shown below.



The CV errors are almost decreasing all the time. We also perform best subset selection, and plot its BIC and adjusted R-square as following.

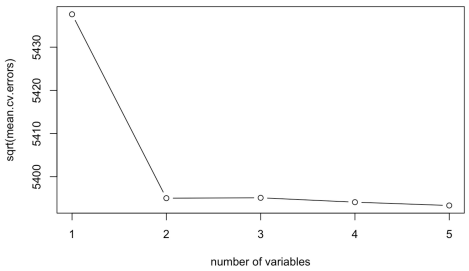




We find that the CV error and BIC are decreasing almost all the time, and the adjusted R-square is increasing almost all the time. All these results reveals that the model might not be unbiased or overfitting; on the contrary, it might be highly biased and underfitting, as the adjusted R-square is not very large. So, the next step is to see whether this model is underfitting.

4.3 Remedying Underfitting: Higher Order Terms

The detection for underfitting is that the training error and testing error are both very high, so we safely infer that this model is underfitting. We can address it by increasing model complexity (adding more features and increasing orders of features). We try to increase the order of features, and then use cross-validation to determine the optimal orders. Note that categorical variables dont need order-increasing, as doing so has no meaning for categorical variables. The CV errors are below.



We observe that using 5th order function might improve the model performance. But since the higher order may be overfitting, we just use the third order to make a gentle model which also gives good prediction.

Their corresponding total error 5395.119, which is the smallest that we have ever achieved. Also, we get an adjusted R^2 of 0.7032. So the model we tuned now does a better job in predicting problem.

5 Tree-based Algorithms: Results and Confidence Analysis

We applied tree-based methods to analyze the dataset. From single decision tree, we further employed Random Forest and various boosting methods to improve the accuracy.

5.1 Single Regression Tree

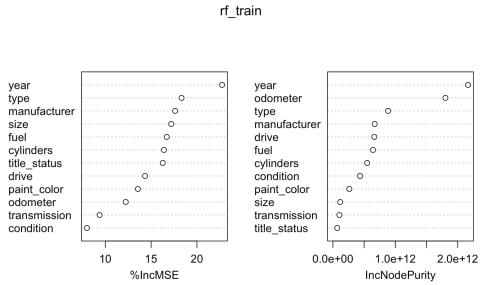
```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 98886 9.669839e+12 11800.360
   2) year< 2013.5 73717 3.235554e+12  8752.353
     4) type=SUV,convertible,coupe,hatchback,mini-van,other,sedan,van,wagon
54245 1.456205e+12  7085.544
       8) year< 2009.5 34713 5.650430e+11  5428.872 *
       9) year>=2009.5 19532 6.265688e+11 10029.840
        18) drive=fwd 9990 1.017362e+11  7407.863 *
        19) drive=4wd,rwd 9542 3.842501e+11 12774.930 *
     5) type=bus,offroad,pickup,truck 19472 1.208805e+12 13395.740
      10) year< 2008.5 11655 4.691575e+11 10111.060
        20) fuel=electric,gas,hybrid,other 8576 1.813163e+11  8041.184 *
        21) fuel=diesel 3079 1.487574e+11 15876.330 *
      11) year>=2008.5 7817 4.264137e+11 18293.140
        22) fuel=gas,hybrid,other 6380 2.140153e+11 16391.450 *
        23) fuel=diesel 1437 8.688735e+10 26736.260 *
   3) year>=2013.5 25169 3.743566e+12 20727.610
     6) cylinders=10 cylinders,3 cylinders,4 cylinders,5 cylinders 10528
4.977517e+11 13199.210 *
     7) cylinders=12 cylinders,6 cylinders,8 cylinders,other 14641 2.220053e+12
26141.100
      14) fuel=electric,gas,hybrid,other 12784 1.609480e+12 24243.520
        28) year< 2015.5 6548 5.755011e+11 20714.990 *
        29) year>=2015.5 6236 8.668479e+11 27948.590
          58) city=FL,LA 644 1.678764e+11 15074.930 *
          59)
city=AK,AL,AR,AZ,CA,CH,CO,CT,DC,DE,DL,GA,HI,IA,ID,IL,IN,JI,KS,KY,MA,MD,ME,MI,MN,MO
 5592 5.799489e+11 29431.180 *
      15) fuel=diesel 1857 2.476387e+11 39204.490 *
```

As a large tree is likely to lead to overfitting, we performed a 5-fold cross-validation to average the tree. The averaged tree has the structure as above, with the root mean square error 6158.212. Its consistent that the prediction of single decision tree is relatively not accurate, and we use it as the benchmark in our tree-based methods.

5.2 Random Forest

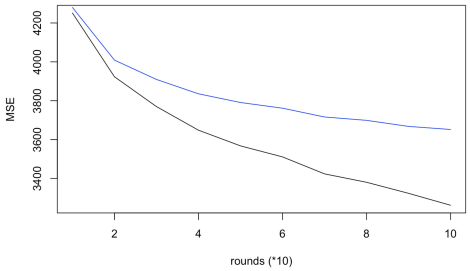
Random Forest can reduce high variance of a single decision tree by averaging the results of bootstrapped data and use a subset of predictors in each tree. We chose $m = \sqrt{p} = \sqrt{16} = 4$ as the number of predictors in each split and ntree = 500 as a large number of trees will not cause overfitting problem in Random Forest. The root mean square error is 4165.886, with variance explained 82.07%. The importance plot below shows that year is the most important variable in the prediction, and type and odometer are also important in MSE and node purity respectively.



5.3 XGBoost

Instead of training on the remaining errors of the strong learner like gradient boosting, adaboost adds the specified week learners to form a strong learner. In practice, we split 80% of the dataset randomly into training set and the other 20% into test set. From the training set, we choose the training rounds from 10 to 100, adding 10 each time. The result of MSE below shows that the XGBoost method is very strong and gives a strong prediction on both training and test dataset, where black line is the training root MSE and blue line is the test root MSE. We can also infer from the plot that if we still increase number of training rounds then the model will

be overfitting.



6 Data-driven Decision Making

We are confident in our results in making decisions. Firstly, we try many methods in linear regression and the tree family, and test for various approaches to develop various models. Based on these models, we are confident in saying that the relationship between price and other features exists and is strong, and we succeed in capturing this relationship using different models. Also, based on the error of the test set and the adjusted R^2, as well as the variance explained, we are confident that our model not only captures the relationship quite well, but so generalizes well on the test data. Thirdly, since our linear regression model is a little underfitting, we would like to get more features if possible to make the model fit more well; for the tree-based models, we detect that the model just fits well, neither overfitting nor underfitting, based on the low error of both training error and validation error.

7 Weapon of Math Destruction

In this project, one concern with our estimates is the potential endogeneity in the asking prices (our labeling prices), since we do not know whether the strike prices are and they could be largely different from the asking prices on adverts. Indeed, cars with better individual specific quality can be sold more easily and, at the same time, be listed with a higher price by dealers. Therefore, the estimated price coefficient can be biased in the positive direction, resulting in underestimated price and vice verse. Correcting the bias in the estimate of the price coefficient is critical for us to assess the importance of each individual feature' effect on the price. Hence, we first cluster data points depending on their major features such as year, maker, mileage and etc. into groups. Then smoothing the data points' asking prices (y-labels) in all quantiles to eliminate outliers and centralize the majority.

8 Fairness

For this project, we handle the potential source of unfairness in machine learning arising from the algorithms. We address specific limitations of p-values or R^2 for high-dimensional stepwise regression in our models through considering a group-level fairness metric. In practice, the most influential statistical criterion appears to be whether a given variable improves performance, as measured by p-values or R^2 statistic. Simply using a p-values or R^2 as the statistical criterion to decide variable inclusion in the prediction function can lead to mistakes, in the sense that improving fit at the person level may not improve fit in the ways that may have even more impact. Inspired by [10], we proposes a fair regression method along with evaluating it with a measure introduced as the "price of fairness" to measure accuracy-fairness trade-offs. The fairness penalties are:

Individual Fairness: for every cross pair $(x, y) \in S_1, (x, y) \in S_2$, a model w is penalized for how differently it treats x and $x^{'}$, weighted by a function of $|yy|$) where S_1 and S_2 are different groups from the sampled population. Formally, this is defined as

$$f_1(w, S) = \frac{1}{n_1 n_2} \sum_{\substack{(x_i,y_i)\in S_1 \\ (x_j,y_j)\in S_2}} d(y_i, y_j)(w.x_i - w.x_j)^2$$

Group Fairness: two groups instances should have similar labels weighted by the nearness of

the labels of the instances. It is defined as

$$f_2(w,S) = (\frac{1}{n_1 n_2} \sum_{\substack{(x_i,y_i)\in S_1 \\ (x_j,y_j)\in S_2}} d(y_i,y_j)(w.x_i - w.x_j))^2$$

Hybrid Fairness: both positive and both negatively labeled cross pairs to be treated similarly over the two groups.

$$f_3(w,S) = (\sum_{\substack{(x_i,y_i)\in S_1 \\ (x_j,y_j)\in S_2 \\ y_i=y_j=1}} \frac{d(y_i,y_j)(w.x_i - w.x_j)}{n_{1,1}n_{2,1}})^2 + (\sum_{\substack{(x_i,y_i)\in S_1 \\ (x_j,y_j)\in S_2 \\ y_i=y_j=-1}} \frac{d(y_i,y_j)(w.x_i - w.x_j)}{n_{1,-1}n_{2,-1}})^2$$

In addition to this method, we also consider the fair regression problem formulation with regards to two notions of fairness statistical parity and bounded group loss[11].

References

[1] S. Singh, B. T. Ratchford, and A. Prasad. *Offline and online search in used durables markets*. Journal of Retailing, vol. 90, no. 3, pp. 301 320, 2014.

[2] A. Jerenz. *Survival analysis: Estimation of the price-response function*. Revenue Management and Survival Analysis in the Automobile Industry.

[3] F. Zettelmeyer, F. S. Morton, and J. Silva-Risso. *How the internet lowers prices: Evidence from matched survey and automobile transaction data*. Journal of Marketing Research, vol. 43, no. 2, pp. 168181, May 2006.

[4] F. S. Morton, F. Zettelmeyer, and J. Silva-Risso. *Internet car retailing*. The Journal of Industrial Economics, vol. 49, no. 4, pp. pp. 501519, 2001.

[5] J. Odink and E. van Imhoff. *Prices of used cars in west-germany before and after the second energy crisis*. De Economist, vol. 130, no. 3, pp. 381396, 1982.

[6] P. Kooreman and M. Haan. *Price anomalies in the used car market*. De Economist, vol. 154, no. 1, pp. 4162, 2006.

[7] J.-D. Wu, C.-C. Hsu, and H.-C. Chen. *An expert system of price forecasting for used cars using adaptive neurofuzzy inference*. Expert Systems with Applications, vol. 36, no. 4, pp. 7809 7817, 2009.

[8] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daume III, Miro Dudk, and Hanna Wallach *Improving fairness in machine learning systems: What do industry practitioners need?* . arXiv preprint arXiv:1812.05239, 2018.

[9] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. *Man is to computer programmer as woman is to homemaker? debiasing word embeddings*. In Advances in neural information processing systems, pages 43494357, 2016.

[10] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. *A Convex Framework for Fair Regression*. arXiv:cs.LG/1706.02409, 2017.

[11] Alekh Agarwal, Miroslav Dudik, and Zhiwei Steven Wu. *Fair Regression: Quantitative Definitions and Reduction- Based Algorithms*. In International Conference on Machine Learning. 120129.

[12] Tianqi Chen, Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System*. KDD 2016, 785-794