Predicting Used Car Prices

ECON 573 Group 6

Jemin Park, Jonah Selom, Keegan McDowell

4/30/2022

Contents

1 Introduction

Filler paragraph 1

text

Filler para 2

2 Data

The data used in this paper was found on Kaggle posted by a user who had originally taken the data from a larger set that included separate datasets for each individual car manufacturer. The original dataset was scraped from thousands as used car listings and was used to predict the selling price of a car that a friend of the creator was debating whether or not to sell. The creator then reshaped the product in order to generalize it more as a prediction model for used cars selling in the UK. The variables used, in this case in an attempt to predict the price of the vehicle, are transmission, mileage, fuel type, tax, miles per gallon, and engine size. The user who transformed the dataset into the one used in this paper was able to clean the data a bit more thoroughly, accounting for any issues made during the original creators scraping process so that the data would then be easier to use. They also changed the breakdown of the data, combining each of the manufacturers and instead split them into predetermined test and training sets, eliminating the randomness of that result.

2.1 Data Preprocessing

The data originally was divided into sections of x_test, x_train, y_test, and y_train. Knowing that some models would need the more complete dataset in order to run, the train data and the test data were both merged by their common column, which was the carID variable, being used as a common identifier between the split data. After that,

there were two variables, transmission, showing whether the car was automatic, manual, or somewhere in between, and fuelType, including both types of gasoline, but also if the car is electric or a hybrid, that could be more useful as factor variables. It was also necessary to examine whether the data had a large amount of missing data, but it was relatively clean when taken from the source, so this was not a major issue. With these barriers out of the way, the data became easier to use through the regression methods used throughout the rest of the paper.

A potential issue with the data would come in trying to run the regression with the car models attribute involved. With 90 different car models, compared to having nearly 7500 total observations, the flexibility of the model becomes a question. Including the variable could lead the model to be too flexible, especially with a low number of observations (suprisingly, a Toyota Camry, one of the most popular used cars, has only 8 observations) causing it to follow any noise too closely and increases variance. A high flexibility can be good when looking solely for prediction, but can cause issues if it is controlling the results of the model too greatly. Collinearity could also be an issue between factors such as mileage and miles per gallon which are likely to decrease/increase as the years go on, but this should be less of an issue because, although they are related in that way, they will still be affecting the price of the car independently.

2.2 Data Visualization

(each into separate sections as needed)

3 Methods

We are interested in building predictive models that accurately determine price based on the aforementioned covariates. While there exist the issues above, we believe for the purposes of this paper and for simplicity, we want to build strong, simple models that can best infer car prices. To do this, we will need to consider outcomes in which we can sufficiently compare test outcomes.

Fortunately, data was split beforehand (as part of a competition dataset), with 4960 observations in the training set, and 2672 in the test set. One key element of interest would be to observe how estimated test errors (in our case, we will use cross-validation error), and how they place may differ from modeled outcomes when calculating test error on the actual test set. For this paper, we will focus on the implementation and predictive ability of 5 key models: linear regression, LASSO regression, ridge regression, boosting, and random forests.

3.1 Linear Regression

We started building a simple multilinear model by regressing all predictive covariates *except* the car model on price. We first excluded `model` to give us an interpretable predictive model, and we found that nearly all the predictors except `fuelTypeElectric`, `fuelTypeOther`, and `transmissionOther` were significant at 0.001% (with the "other" fuel type being significant at 0.01%), with an impressive R^2 of 0.7194. We note that every brand listed has some significant predictability on their respective effects on price. However, upon examining diagnostic data, we see that the standardized residuals indicate irregularity in the upper quantiles.

So, we then tested a base model on the log of car price, we found that this addressed the issue of regularity in the model, while simultaneously increasing the

significance of the predictors. This had improved the R^2 to 0.884 on the base model that we were testing. Thus, continuing forward, we will take log price into full consideration for all following models. Below is the summary output for the log model on all predictors but the car model:

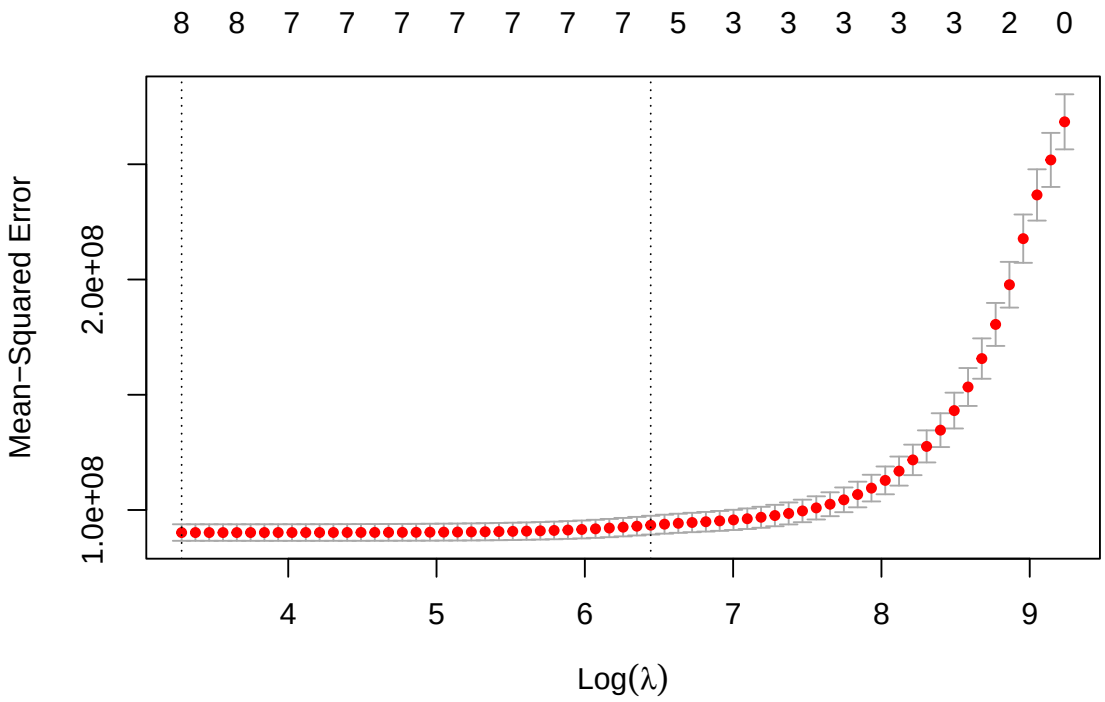
```
##
## Call:
## lm(formula = train.lprice ~ . - model - price, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28268 -0.14979 -0.00331  0.14194  1.61470
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.410e+02  3.889e+00 -61.971  < 2e-16 ***
## brandbmw              -4.015e-02  1.611e-02  -2.492  0.01274 *
## brandford             -4.009e-01  1.804e-02 -22.223  < 2e-16 ***
## brandhyundi           -5.207e-01  1.866e-02 -27.905  < 2e-16 ***
## brandmerc             -1.007e-01  1.591e-02  -6.330 2.67e-10 ***
## brandskoda            -5.032e-01  2.068e-02 -24.330  < 2e-16 ***
## brandtoyota           -3.958e-01  2.000e-02 -19.796  < 2e-16 ***
## brandvauxhall         -5.922e-01  2.235e-02 -26.497  < 2e-16 ***
## brandvw               -2.839e-01  1.700e-02 -16.696  < 2e-16 ***
## year                   1.243e-01  1.926e-03  64.554  < 2e-16 ***
## transmissionManual    -1.222e-01  1.017e-02 -12.015  < 2e-16 ***
## transmissionOther     -1.626e-02  2.362e-01  -0.069  0.94513
```

```
## transmissionSemi-Auto  4.765e-02  9.282e-03   5.134 2.95e-07 ***
## mileage               -5.495e-06  2.237e-07 -24.567  < 2e-16 ***
## fuelTypeElectric       5.667e-01  1.750e-01   3.239  0.00121 **
## fuelTypeHybrid         3.619e-01  2.545e-02  14.225  < 2e-16 ***
## fuelTypeOther          2.724e-01  5.072e-02   5.372 8.15e-08 ***
## fuelTypePetrol        -9.726e-02  8.172e-03 -11.902  < 2e-16 ***
## tax                    8.936e-04  5.068e-05  17.631  < 2e-16 ***
## mpg                   -5.552e-04  1.337e-04  -4.152 3.36e-05 ***
## engineSize             2.527e-01  6.907e-03  36.590  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2358 on 4939 degrees of freedom
## Multiple R-squared:  0.884,  Adjusted R-squared:  0.8835
## F-statistic:  1882 on 20 and 4939 DF,  p-value: < 2.2e-16
```

Next, we want

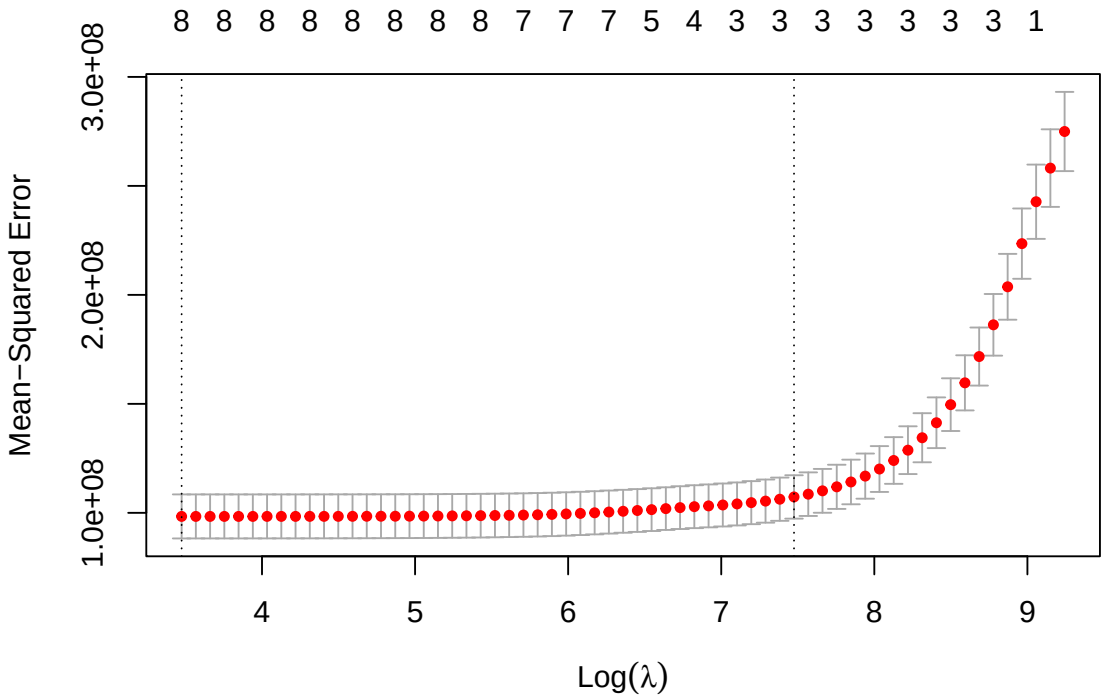
3.2 LASSO Regression

```
## [1] 26.60289
```




```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -3.306191e+06
## carID         .
## year          1.639614e+03
## transmission  1.744855e+02
## mileage      -1.435038e-01
## fuelType      1.587856e+01
## tax           .
## mpg           6.456150e+00
## engineSize    1.215392e+04

## [1] 9665.45
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)  -3.535277e+06
## carID         1.255783e-02
## year          1.750376e+03
## transmission  5.689725e+02
## mileage      -1.482919e-01
## fuelType      5.081973e+02
## tax          -1.105825e+01
## mpg           3.275726e+01
## engineSize    1.398212e+04
## [1] 0.6673753
## [1] 32.2968
```

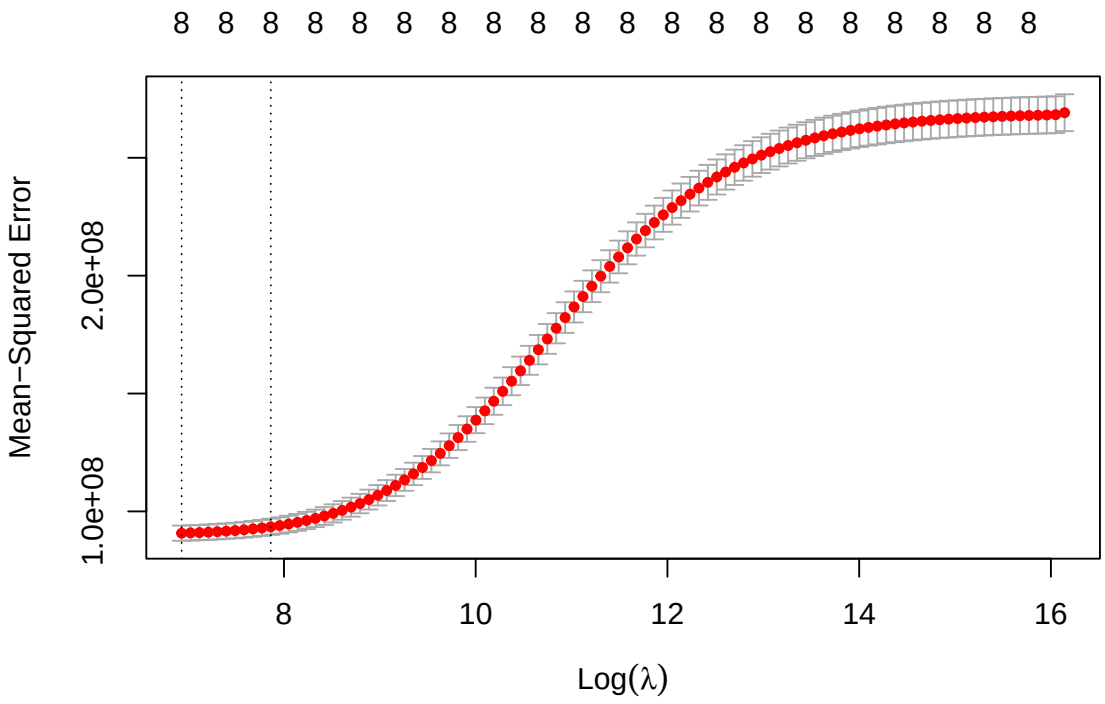


```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                           s1
## (Intercept)  -1.998530e+06
## carID         .
## year          9.936637e+02
## transmission  .
## mileage      -1.792539e-01
## fuelType      .
## tax           .
## mpg           .
## engineSize    1.067156e+04

## [1] 10358.71

## 9 x 1 sparse Matrix of class "dgCMatrix"
##                           s0
## (Intercept)  -2.538933e+06
## carID         1.000079e-01
## year          1.256280e+03
## transmission  6.818544e+02
## mileage      -2.165250e-01
## fuelType      4.504767e+02
## tax          -9.153155e+00
## mpg           3.943719e+01
## engineSize    1.392701e+04

## [1] 0.6527993
```

3.3 Ridge Regression

Discussion & results

```
## [1] 1025.15
```




```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -3.110045e+06
## carID         2.925588e-02
## year          1.542003e+03
## transmission  7.158268e+02
## mileage      -1.480873e-01
## fuelType      2.529720e+02
## tax          -2.872662e+00
```

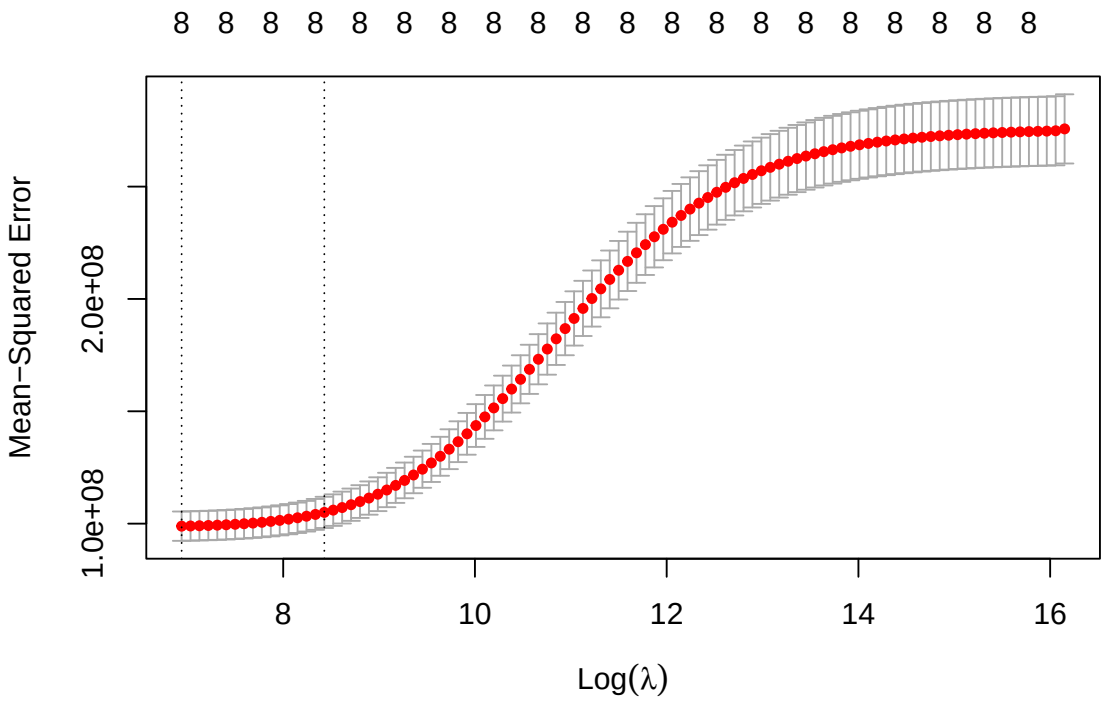
```
## mpg           1.673819e+01
## engineSize    1.144305e+04
```

```
## [1] 9668.032
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)  -3.342459e+06
## carID         2.702894e-02
## year          1.655784e+03
## transmission  6.513949e+02
## mileage      -1.499740e-01
## fuelType      3.939827e+02
## tax          -7.321506e+00
## mpg           2.596203e+01
## engineSize    1.287057e+04
```

```
## [1] 0.6647528
```

```
## [1] 1033.261
```




```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -2.496825e+06
## carID         1.138448e-01
## year          1.238842e+03
## transmission  8.308350e+02
## mileage      -1.776628e-01
## fuelType      1.985546e+02
## tax           1.030000e+00
## mpg           1.218882e+01
## engineSize    1.016214e+04
```

```
## [1] 10247.38
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)  -2.586542e+06
## carID         1.161266e-01
## year          1.280692e+03
## transmission  7.522950e+02
## mileage      -2.048293e-01
## fuelType      3.850361e+02
## tax          -5.904064e+00
## mpg           3.180265e+01
## engineSize    1.285808e+04
```

```
## [1] 0.6504537
```

3.4 Model 4

Discussion & results

3.5 Model 5

Discussion & results

4 Conclusion

Final discussion with interpretation from model, inferences, best results, etc.