Used Car Pricing

Spencer Halverson, Connor Pearce, Daniel Mortenson, Evan Nuss

Description

We want to see how accurately a machine learning model can predict the price of a used car based on certain characteristics of the car. This model could be used to predict the cost of future models of the same car to determine if it's worth waiting for a newer model.

There's also the question of fair car pricing. Many times, individuals or dealerships will overprice a car to make a greater profit. Different car buying sites will often give a rating of the price fairness. However, they don't give any comparison data or say what that rating is based on exactly. This model can take the uncertainty out of the validity of such ratings. We'll be able to run the model on a given car we are interested in buying to determine if it is fairly priced.

We hope the model will pinpoint the features that most affect a used car's price, allowing us to understand how much a certain feature will boost the price and decide whether that feature is worth it.

Data Gathering

We initially tried to web scrape some data from websites such as Kelley Blue Book, but decided it was not prudent given our time constraints. Therefore, we are using previously gathered data available on Kaggle, which we have cleaned and are now figuring out which features are most important to include. More details on the dataset will be given below.

Data Set

Our dataset has 458,213 samples. Our data initially had 25 features: `'id', 'url', 'region', 'region_url', 'price', 'year', 'manufacturer', 'model', 'condition', 'cylinders', 'fuel', 'odometer', 'title_status', 'transmission', 'VIN', 'drive', 'size', 'type', 'paint_color', 'image_url', 'description', 'state', 'lat', 'long', 'posting_date'`.
Of these, only 6 were numerical: `'id', 'price', 'year', 'odometer', 'lat', 'long'`.
Some of these features have unique values for each sample (such as `'id'` and `'description'`) so we decided to drop those features. Additionally, many of our data samples were incomplete (i.e., had features with missing values). Therefore we needed to handle missing data as well as categorical data.

Missing Data

We first tried the naive approach for dealing with missing data: dropping all samples that had features with missing values. However, this left us with only 42,384/458,213 samples, or 9.2% of our data. Instead, we decided to replace numerical NaNs with the median value for that feature.

For categorical data with missing values, we replaced NaNs with the string `'none'`. This way, whichever model we apply to the data can learn to deal with the missing data itself.

Categorical Data
Some categorical features had only a few attributes (for example, `'drive'` had only 4 categories: `'fwd','4wd','bwd'` and `'none'`) but others had many more (for example, `'state'` had 51 attributes). One approach we tried was using the `pandas` function `get_dummies()` to one-hot encode each categorical feature so that each attribute was in its own dimension. However, to reduce complexity, we decided to drop any features with more than a certain number of attributes (this number can change depending on the model used). When dropping any feature with 10 or more attributes, we are left with 47 features total (counting each categorical attribute as an individual feature).

Another approach we discussed was partitioning categorical data into sub-categories. For example, there are more than 40 distinct values for `'manufacturer'` in the data, but Toyota, Honda and Chevrolet make up the vast majority of them. By taking the few most common manufacturers as distinct attributes and lumping all the other ones together as `'other'`, we can vastly reduce dimensionality while still preserving important distinctions in a given category.

Here is a sample point (pre-processed):

```
7240110508
'https://oklahomacity.craigslist.org/ctd/d/noble-2016-ford-f350-crew-cab-f
latbed/7240110508.html'
 'oklahoma city'
'https://oklahomacity.craigslist.org' 28900.0 2016.0
 'ford' 'f350' 'excellent' '8 cylinders' 'gas' 78000.0 'clean' 'automatic'
 'none' '4wd' 'full-size' 'truck' 'silver'
 'https://images.craigslist.org/00Q0Q_8ZrD6U3chFs_0ww0oo_600x450.jpg'
 'CUSTOM AUTO & EQUIPMENT WWW.CUSTOMAUTOEQUIPMENT.COM 2016 FORD F-350 CREW
CAB DUALLY LONGBED, 6.2L V8 GAS IN AUTOMATIC AND 4WD, XLT PACKAGE WITH
POWER WINDOWS AND LOCKS, BLUETOOTH, VINYL FLOOR AND CLOTH SEATS, HAS A CM
FLATBED WITH SIDE TOOLBOXES, TRUCK RUNS GREAT CONTACT JD OR JUSTIN
405-351-0884 405-834-4786'
 'ok' 35.123808000000004 -97.37087 '2020-12-01T16:05:19-0600'
```

Models

After we gathered and cleaned up the data, we immediately put the data into a simple Decision Tree to see a baseline accuracy for the output. The output provided a median absolute error of $700. Going forward, we will focus our efforts in optimizing this median absolute error using 2 different approaches. Spencer and Evan will continue optimizing a decision tree model and Daniel and Connor will focus on creating and optimizing a random forest model.

Schedule and Plans

Moving forward, we plan on finalizing our models by March 25. Next we will have a draft for the paper and script for the video presentation by April 1. We will then finish our video presentation and be mostly done with the paper by April 8. Finally, we will complete the paper and submit both the paper and video presentation on or before the due date, April 13.