OTD Depreciation Project Summary

Ahmed Awadalla

2023-11-10

Project Summary

In this project we will determine the factors affecting vehicle price for cars and trucks in Florida. We will also develop a live gradient boosting model to predict the market value of any vehicle.

Data Procurement and Analysis

- We will aim to scraped and cleaned 100,000 Craigslist vehicle listings from Florida.
- We will then train a gradient boosting model to predict the market value of vehicles, I have already written the code for this part, all we need to do is input the data.
- Then train a linear model to predict a dollar value for important vehicle features (ie. size, fuel type, manufacturer, odometer, etc.)
- For example, Predictions indicate that diesel vehicles are worth 9,000 dollars more than non-diesel vehicles on average.

Exploratory Data Analysis:

- From a smaller dataset, I found depreciation time constants for the most common vehicle models and manufacturers (ie. which vehicles 'hold their value' the best). Found Toyota, Honda, and Volkswagen depreciate the slowest, while Dodge, Chevrolet, and Ford depreciate the fastest.
- If needed we can use interpolation/smoothing to create average contours of price vs odometer reading and age. This will give a benchmark depreciation for the state of Florida only. For example, benchmark depreciation of $0.20/mile driven for the region (Florida).

Actions Steps:

1. Web Scraping

I used this tutorial https://towardsdatascience.com/web-scraping-craigslist-a-complete-tutorial-c41cea4f4981 as a basis to scrape vehicle listings from Craigslist. I need to expand on the code a fair bit in order to scrape data from each individual vehicle listing page (as apposed to the search page which displays 120 listings at a time). In the end I was able to extract 16 features from each listing:

body text (the main text of the listing) condition (eg. new, salvage) drive (front, rear, or 4wd) fuel (type of fuel) latlong (latitude and longitude associated with the posting) location (location associated with the posting, ie. 'Florida') make (year make and model of the vehicle in a string, ie. '2020 Toyota Corolla') odometer (odometer reading in km) paint color price (in $ USD) sale type (owner or dealer) title status (ie. clean, rebuilt) transmission (manual or automatic) type (ie. sedan, minivan) cylinders (number of cylinders in the vehicles engine) size (ie. full-size, compact)

2. Data Cleaning

About 11% of the scraped data was missing values from the smaller dataset. The columns with missing values were:

condition: 21.3% drive: 24.5% odometer: 0.3% paint color:29.8% type: 30.7% cylinders: 37.4% size: 57.8%

The person who scraped this data opted to fill most of this missing data before going forward. 'type', 'size', 'drive', and 'cylinders' were filled with the mode of the data, 'odometer' was filled by the mean, and 'paint color' and 'condition' were replaced with 'unknown'. We will try to take a similar approach when cleaning our data

Additional cleaning steps were:

Combine 'pickup' and 'truck' into one category Convert the 'price' from a string to a number Split the latitude/longitde into distinct numerical columns Clean up the location column by taking only the first word of the location with some exceptions Split out the year, make and model of the vehicle into separate columns Split out the newly created model column into a base model and modifiers (many models have modifiers at the end such as 'lxt') Create an 'age' column by subtracting the current year from the 'year' column Create new columns based on if the text body contained words I selected as 'positive' (ie. 'vintage') or 'negative' (ie. 'torn').

3. Model Building

Will will try to fit a tree-based gradient boosting model. Only minor changes to the data have to be performed before fitting: outliers with a price above $100,000 and odometer above 600,000 miles must be dropped, and the few rows with missing data should be dropped. LightGBM can handle categorical features directly so no encoding is required.

Optimal hyperparameters for the model can be found with a cross-validated grid-search over the entire dataset (sklearn GridSearchCV). Next 20% of the data was should be held out for validation.

Finally we will construct an EDA report with visuals that developers can use on the site.

Example of great depreciation calculator.

https://caredge.com/depreciation