DDR Final Project-Used Car Image and Price Dataset

Jake Brophy, Akshay Prasath Manickavasagam, Arjun Velmurugan Mahesh

Table of Contents

Executive Summary

Used cars represent a huge market in the US, with companies like CarMax, CarFax, and Cars.com all getting much or all of their business from this market. The goal of this project is to put together a business-ready dataset that can be used to develop models for predicting the fair price of a used vehicle. The main business purpose of this would be to help inform customer decisions on whether or not to purchase a vehicle by providing them with an accurate estimate of a vehicle's true value. Many vendors such as CarFax already provide something like this to their customers, but Craigslist does not despite being a large hub for people seeking to purchase or sell used vehicles. Craigslist is not specifically a website dedicated to the sale of used cars, but is instead a classified advertisements website that many people use to sell and purchase vehicles, which is why it was chosen as the data source for this project. It was also chosen because many typical used car websites have strong blockers in place to prevent people from web-scraping them, whereas Craigslist does not. Three regions of California were chosen for scraping: Los Angeles County, San Diego County, and the Bay Area. In total, roughly 20,000 used car listings were scraped, along with 17 different features about the car being listed for sale and the post itself. BeautifulSoup and Selenium were used for the web-scraping section, RegEx was used to format all fields to minimize the cleaning needed in any analysis projects, and MongoDB was used as the database due to the sparse nature of the dataset. A mix between a flat and nested structure was used to structure every document. This was used in order to strike a balance between being able to load and flatten the data for machine learning projects and allowing for optimized querying of fields. We provide more details on the background, business value, technologies used, and the choices made in database structure below.

Background, Context, and Domain Knowledge

The purchase and sale of used cars is a large market in the US, with companies like CarFax, CarMax, and Cars.com all having large stakes in this market. A big part of the customer experience for these companies is a predicted car price generated from the car's history, which potential buyers can use to make decisions about whether a car price seems fair or not. Take CarFax for example, which specifically has something called CARFAX Value, which they describe as a fair price that they generate from a car's history (CarFax, 2023). We plan on investigating this same business problem, but for Craigslist, a company which has a wealth of information of used cars, but has not yet leveraged this to create business value. We collect data on used car prices as well as all of the car features available on Craigslist in order to put together a dataset that can be used for future research on how to best determine a car's price from its history and features.

Introduction to data sources

Craigslist is a classified advertisements website where people list jobs, items for sale, as well as used cars, which is what we were interested in for this project. 3 separate regions of California were scraped: San Diego County, Los Angeles County, and the Bay Area, with the Bay Area being the largest of the 3 counties followed by LA and then SD.

Selenium and BeautifulSoup were both used to access, download, and parse the required pages. Because Cragislist requires javascript to be enabled for the website to function, we found it difficult to scrape the full page using BeautifulSoup. As a result, we used Selenium to open the

page in chrome, download the full page to disk, and then continuously moved to the next page until a certain number of pages had been exhausted and downloaded to the appropriate directory. From there, we found that BeautifulSoup was able to effectively access each individual listing, so we parsed through the downloaded files with BeautifulSoup, accessed each individual listing, parsed each car feature that we wanted to use for analysis, and downloaded the individual listing for later use. The features that were retrieved from every page are as follows:

1. Full name of the car (i.e. 2008 Ford Fusion)
2. Brand (i.e. Ford)
3. Year (i.e. 2008)
4. Fuel (gas, electric, or diesel)
5. Odometer (number of miles on the vehicle)
6. Condition (Good, Fair, or Bad)
7. Paint color
8. Size (i.e. full, compact, etc.)
9. Number of Cylinders
10. Title status of vehicle (has it ever been severely damaged)
11. Transmission (manual or automatic)
12. Drive (4wd vs Front wheel drive)
13. Body type (SUV, sedan, etc.)
14. Image url for later use in image classification
15. Price
16. Date of post
17. Date the post was updated

For the year and brand factors, those were not separately listed fields within Craigslist so RegEx was applied to the full car title to get the year, and then the full title was parsed to find if it contained 1 of 24 car brands which was then passed into the brand object. For almost all of the other objects, when the raw values were scraped from Craigslist they had a structure similar to "Price: 11000" or "Condition: good" so the Python replace function was used to get rid of these extra words before being passed into the database. The numeric variables year, odometer, number of cylinders, and price were all converted to a float value, and the two date values were both converted to datetime objects.

Roughly 20,000 used car listings were scraped across all 3 cities. However, due to the fact that Craigslist does not require people to include all those values in their posts, many of those fields are missing and using a MySQL database would result in many sparse values that would be a waste of space. As a result MongoDB was chosen because it can handle sparse values very well.

Database Design and Justification

The database contains 3 collections corresponding to each of the 3 regions, an example document is provided below:


```
{
"_id" : ObjectId("641404f459aa7868e84596fd"),
"url" :
"https://sfbay.craigslist.org/sfc/cto/d/san-francisco-honda-element-2008-ex/7596939831.html",
"full_car_brand" : "Honda Element 2008 EX",
"year" : 2008.0,
"brand" : "honda",
"fuel" : "gas",
"odo" : 156900.0,
"cond" : "excellent",
"paint" : "grey",
"cyl" : 4.0,
"title" : "clean",
```

"trans" : "manual",
"drive" : "fwd",
"body_type" : "SUV",
"img_url" : "https://images.craigslist.org/00707_6t1Lg18wmoT_0CI0t2_600x450.jpg",
"price" : 4500.0,
"posted_stats" : {
"posted_time" : ISODate("2023-03-16T21:24:00.000+0000"),
"last_updated_time" : ISODate("2023-03-16T21:24:00.000+0000")
}
}

A largely flat document structure was used, with fields 1-15 all being contained within one document so that it can be easily accessed and converted into a pandas or R data frame for machine learning projects.

Additionally, we wanted to ensure optimized lookup of fields that might be of interest. As a result, we set the variable ‘brand’ as the index, and we created a document within each document that contains when the car was posted to Craigslist and when the post was last updated. Setting the brand as the index will allow faster lookup of these values through binary search, and the posted stats were placed in a nested document mainly because these variables likely don’t have much bearing on the cost of a car. However, they might be useful for querying purposes to find car postings within certain date ranges. The nested structure of the document allows for faster lookup of these fields due to the reduced number of values the system has to search through before finding variables that match whatever the set criteria is.

The resulting structure is a convenient blend of fast lookup of necessary fields with convenience in converting each document into the required format for machine learning and other price analysis projects.

17 variables were chosen because that was the maximum number of allowable fields for a car on Craigslist, and we explicitly chose to take all of them because we wanted as much information as

possible available to us when investigating the question of what affects the price of a used car. Additionally, on Craigslist, all of these features are visible to any user who looks at a listing, so it is likely that all of them are factoring into their willingness to pay for a car. The business value added by this dataset is for companies like Craigslist who want to conduct analyses of what factors are contributing to the price of a car, other companies who want to expand the data available to them to build models for analyzing the fair market value of a used vehicle, or individuals who want to flip cars for a profit and want to find undervalued vehicles.

The img_url object contains the Craigslist url of the first image a user sees when they open a listing. Obviously, in its raw format this is not a helpful determinant of price. However, it can be used to download the images of all 20,000 listings for use in image classification tasks. The usefulness of this in predicting price is varied depending on the goal, but an example could be to use an image classification algorithm to fill in missing key values such as brand or year, both of which are likely heavy determinants of price.

Summary/Conclusion

The main goal of this project was to scrape a used car dataset with the potential to be used as a price prediction and image classification tool for sellers of used cars who want to engage in price optimization. Craigslist was chosen as the site from which we would retrieve our data due to its relative ease of access and abundance of used cars. 15 covariates were scraped from 20,000 listings across 3 major California cities to provide a comprehensive picture of the factors impacting the price of a used car. MongoDB was used to limit the amount of space being taken up by the sparse fields in our database, and a flat document structure with certain fields of

interest optimized for querying was used. This allows us to balance optimized lookup of values with a structure that is easy to flatten in languages like Python for machine learning projects.

Works Cited

"What Is Carfax Value and How Is It Calculated?" *CARFAX*, https://support.carfax.com/article/what-is-carfax-value-and-how-is-it-calculated.

Code Documentation

Below is a list of the files contained in the submitted zip file and what they do

- Craigslist_cars_mongo_dump
    - MongoDB dump of all scraped data
- final_project_ddr_craigslist.ipynb/.py
    - Script for scraping Craigslist Data
- full_page_bay/la/sd
    - Directory containing all the full web pages from Craigslist corresponding to the region they were scraped from
- individual_page_la/sd/bay
    - Directory containing all the individual listings from their respective regions