David Chehet

Programming Assignment I Essay

Within my Google Colab notebook, I constantly jotted notes about my actions and why I was doing them. I will begin this essay by labeling the section of cleaning I was working on, and summarizing my notes, to hopefully make this very easy and straightforward to follow. At the conclusion of the essay, I will address the additional questions listed in the assignment.

I began my data cleaning with handling nulls as this is what I was most comfortable with. The columns which I removed ended up being county and size. I choose to remove these two entirely because their null value percentages were 100% and 70%, respectively. After, I moved to the columns cylinders, condition, paint color, drive, VIN, and type, dealing with their null values by replacing them with a string 'Unknown'. My rationale was that for all of those fields, the reason why they are missing is because they were not filled in, rather than not existing. We learned from the Kaggle course that if the field was simply not entered, it wouldn't be extremely useful to try to 'predict' what it might have been, especially since we have no way to predict that. On reflection, maybe if we had a Google API Key, we could use it to infer cylinders from make/model, and done that same with other fields, but alas, we did not have that liberty. When I started handling nulls, 11.5% of the entire data set was null, and after it was under 1%. I felt this was sufficient to move to the next section, outlier removal.

Outlier removal was quite enjoyable for me, mainly because it was interesting to see how big of an effect they can have on your data. I removed the outliers from fields like year,

price, and odometer. I will briefly explain my work in each one. Year mainly had outliers on the bottom end, with the min being from 1900(must have been an antique vehicle). Regardless, I ended up removing 16000 rows and slicing our standard deviation in half, from 9.5 to 5. Next, price was arguably the most ridiculous. The dataset included 32000 cars listed at $0, and 1 listed for $3.7 billion dollars. The problem I ran into was that when I used the function from year, the price field still was inaccurate. It kept the 32000 free cars and cut off cars above 50K. I felt that there might be some legitimate sales, however, on the higher end, thus I instated a hard-coded upper cap of $100k, and a lower cap of $1 to eliminate free cars and likely scams. Lastly, I removed outliers for odometer by capping them at a 500k upper threshold and keeping 0 in case the cars are in fact new.

Feature creation was relatively simple. I removed year and created an age field, and I created price/miles driven, and miles/year driven to have a better idea of the usage the cards went through and the discount rate per mile driven the customer is looking at.

In feature selection, I dropped a lot of columns that were mainly categorical and would have literally zero use. Columns like description, VIN, region URL, etc. In the notebook where I remove them, I jotted down my notes from the decisions I made, for example how I decided to remove region because state, latitude and long would tell us that. It is here that I will address the description field. At first, I didn't delete it in the feature selection, because I imagined it could be useful. Specifically, if I had the knowledge to do so, I would consider using a machine learning algorithm to detect keywords in the description that might tell us valuable information, like if this was a FSBO (For sale by

owner) or sold by a dealership. However, since that fell outside the scope of this assignment, I did not go through with it.

Feature transformation was a good bit of fun. I got to learn a lot about the 'dummies' method. I ended up doing binarization of title status, mainly because the non-clean entries were a minority, so I figured I would group them as a 0 and make clean status a 1. Then, I mapped the categorical condition column to a discrete ordinal ranking from 1-5. Lastly, I mapped the extremely continuous odometer column to categorical levels of Low, Moderate, High, and Very High. These changes made the data a lot cleaner in my view.

Lastly, for normalization, I normalized my price and age columns. In the first set up graphs, you can see that the normalization looked good for price but not for age, which is why in the second graph, I decided to do scaling for age rather than normalization. The reason why I did this pivot was because of the contrast in bell curve between age and price in the first attempt. Overall, I really enjoyed this assignment and am looking forward to the next one. I like the hands-on approach to learning as it helps me become more comfortable in Python as well.

Lastly, to address the class imbalance issue, I think using a Google API Key would be a solution. For example, if we have the model but not the make of a car, the API will help us match that up. It would be a similar procedure for filling in the details about cylinders, transmissions, drive, etc. Ultimately, if we have the model of a car, we can figure out the rest of the information with Google and drastically remove Unknown/NaN variables.

Citations

- https://www.kaggle.com/code/alexisbcook/scaling-and-normalization
    - I borrowed code from the Kaggle course mainly for the plotting of the graphs for scaling and normalizing.