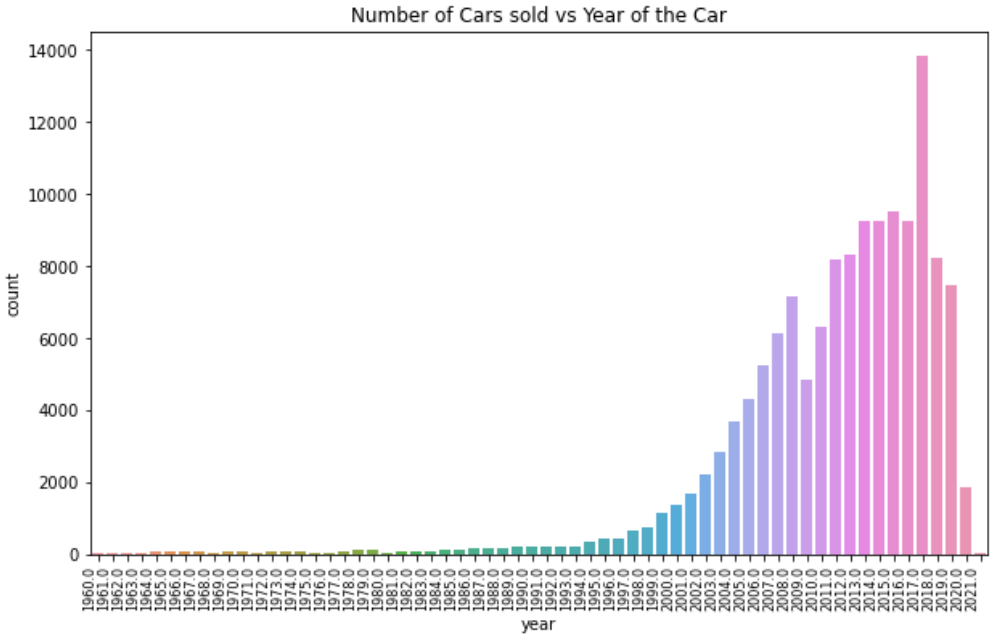
Eric Gleiter
Anyesha Ray
Final Project Part Three
11/10/2020

Data Preparation and Cleaning

Our dataset is a premade dataset that is available from the website kaggle (https://www.kaggle.com/austinreese/craigslist-carstrucks-data?select=vehicles.csv). The data is gathered via a web scraper that pulls information from craiglist every few months. Then the updated data is published to kaggle.

In order to clean our data we first dropped unnecessary data, such as the url of the craigslist post (which is often outdated), the latitude and longitude of the car, (as we have access to location via county and state), and other columns relating to the craigslist post (image_url, region_url, etc…). Then we began to look deeper into the data. We noticed that the vast majority of car sales were from the year 1960 and onward (see plot below). Additionally we reasoned that as the years go by older cars are less and less likely to be for sale. Thus we decided to limit the tail end of the data set to the year 1960.



Next we looked at other variables to identify similar issues. We noticed that a large chunk of data had a price of 0. Upon further investigation we discovered that this was because the seller was looking for the “best offer” and did not set a price. Due to this we removed any row which had a zero for price as we have no way of knowing the

true sale price for that vehicle. Similarly we had issues with missing data many posts simply did not include some details such as county or state of the sale. In order to solve this problem we decided to remove any columns with more than 50% of the data missing. These columns were: price, year, manufacturer, model, fuel, odometer, title_status, transmission, drive, type, paint_color, and state. Our cleaned dataset has 12 variables and over 130,000 data points to work with. Outside of the stated anomalies of 0 prices, we have one potential problem. While we removed columns with an extremely large number of missing variables we still have to deal with missing values in the data set. When we perform feature selection we will have to be careful about how we deal with these variables.

Machine Learning Model Training

After performing feature selection, which in our case was done by selecting the columns with the highest correlation to the price variable. We then split the data into testing and training data. We used an 80/20 split with 80% of the data being for training and 20% being used for testing. In order to improve our features we performed one hot encoding so that we could incorporate the categorical variables into our regression model.

In order to find a model that worked we attempted to train 6 different models: a linear regression, polynomial regression (degree 2-5), regularized polynomial regression (also degree 2-5), a random forest regressor, an XGB regressor, and using ADABOOST. The results for the models are shown in the table below. As you can see the linear model has the lowest RMSE and MSE out of all 6 models. This is the reason that we chose the linear regression model as our final model. Unfortunately, the linear regression model has no hyperparameters for us to go back and tune in order to further optimize the results.

Model	MSE	RMSE
Linear	19931065637.303257	141177.42608966652
Polynomial (Degree 2)	195640166798.01044	442312.2955537303
Regular Poly (degree 2)	195564026683.8936	442226.21664018696
Random Forest	1495321125361.1665	1222833.2369383678
ADABoost	2152539804059.767	1467153.6402366888
XGB	1315958787997.9521	1147152.4693770886

Summary and The Future

By looking at the correlations from the features to the selling price, it was quite clear to us that the model would struggle to accurately predict price. Most correlations were less than .01. This led us to believe that without some more serious effort all models we tried were going to be rather unsuccessful. Should this project be repeated in the future, one clear improvement that could be made is more feature engineering. Due to the lack of time and resources available to this project, we made the simpler choice to remove hard to work with data rather than try to impute or to combine columns in interesting ways. We believe that more effort put into feature engineering would provide the largest return on investment for this project. Finding features that correlate well with the price would improve the model dramatically. Additionally please see the accompanying jupyter notebook for any and all code related to this project.