CAR PRICE PREDICTION

Liana Bezhanyan
Tatevik Poghosyan
Saten Harutyunyan

August 16, 2019

Contents

1 Introduction

Auto.am is a website where you can find cars of various brands that are put into sale and information about them.Our goal is to scrape this website to collect data, clean and translate it, fit a linear model to predict to price variable. The implemented is in python programming language.

2 Scraping and Cleaning the Data

2.1 Scraping

For Scraping the data we used selenium, scrapy and requests packages. Selenium is used because tables is generated by JavaScript. We scraped brand, model, price from year for main pages,also we collect individual URLs of cars. More specific data we got from these individual URLs of each car.

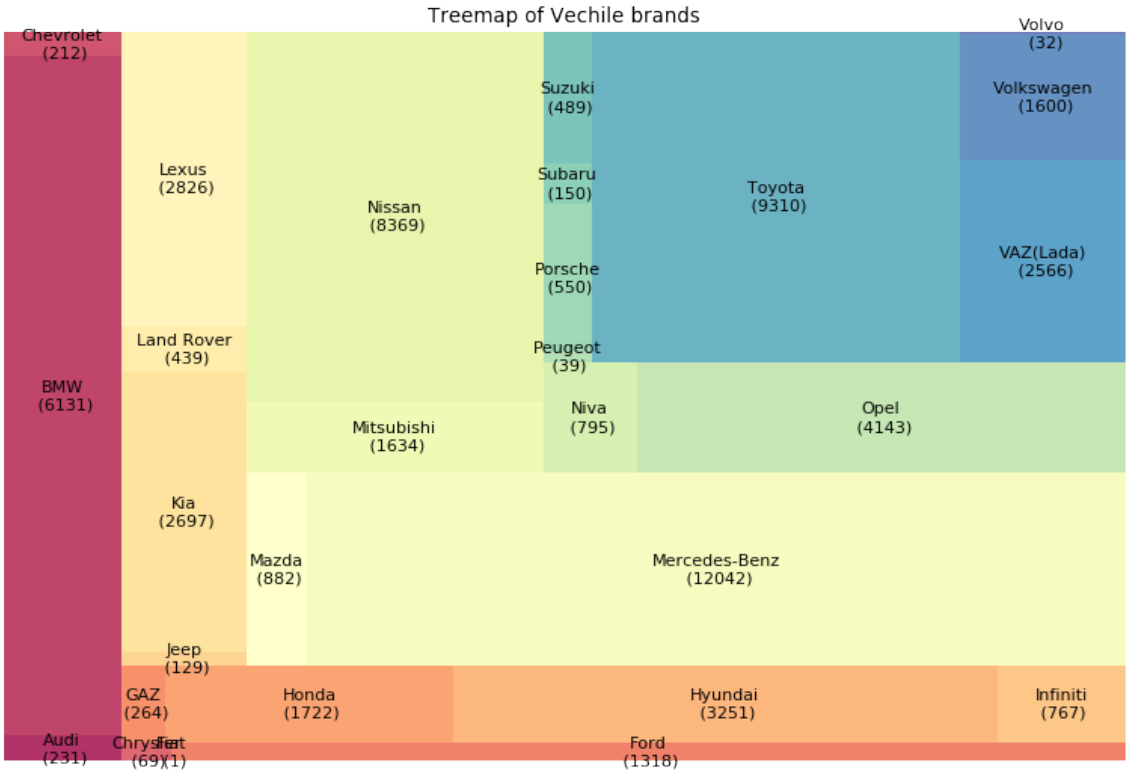
2.2 Cleaning

We wrote these functions for cleaning and reformatting the data for numerical variables: exchange (dram to dollar), mile to km, remove spaces and symbols. In these functions we used regular expressions. Next, we drop the rows that don't contain price variable, as it is crucial for our learning algorithm. After that we dropped all the rows that contain Nan. As this step made our data very scarce, we dropped the variables which contained most nans. These were the number of the doors, modification, the size of the wheels. Our data was in Armenian, for that we used Google's translate API to translate our categorical data. We saved our date frame to a csv file.

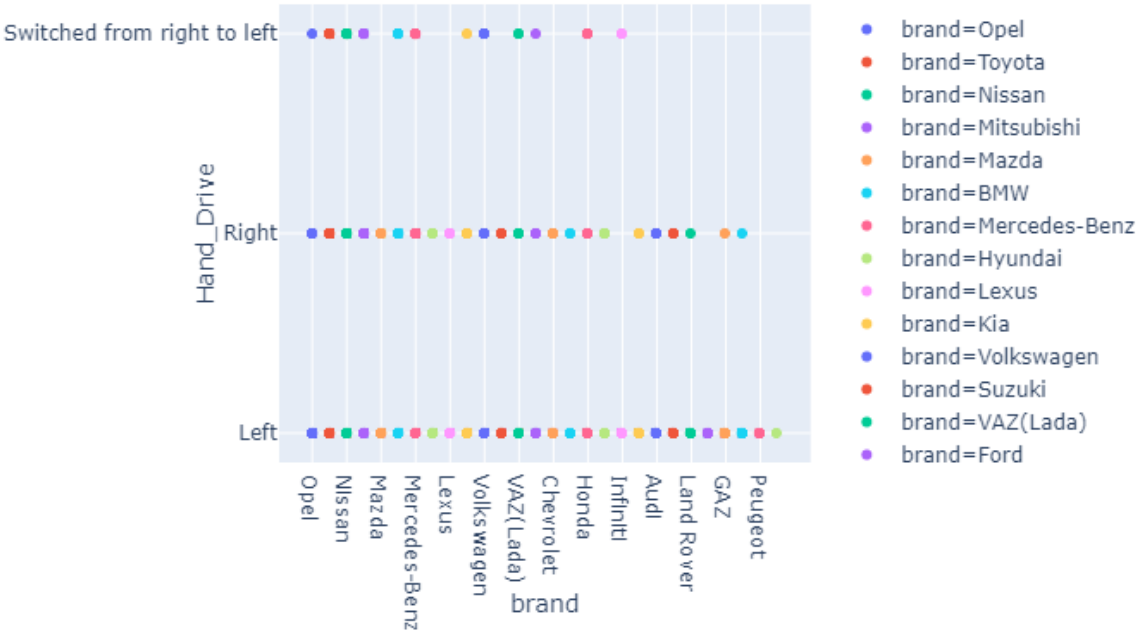
3 Analysis and Linear model

3.1 Some demonstrative plots

We started analysis with a tree map that shows the proportions of each brand, and it is visible that Top 3 car brands are Mercedes Benz , Toyota and Nissan respectively.



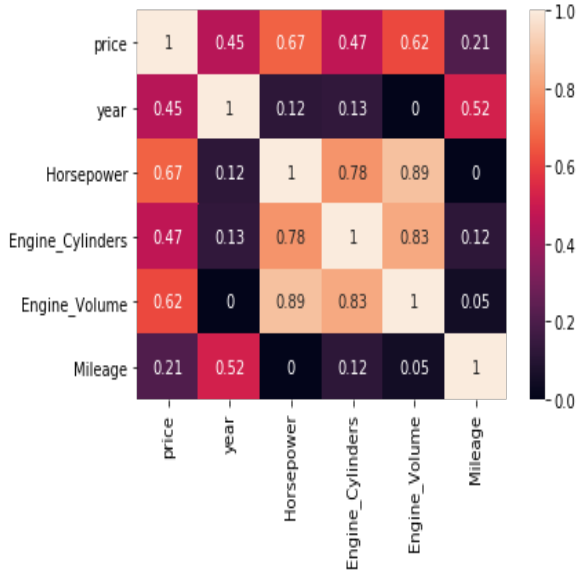
We also made a plot which shows brands with right hand side wheel, left hand side wheel and switched from right to left .



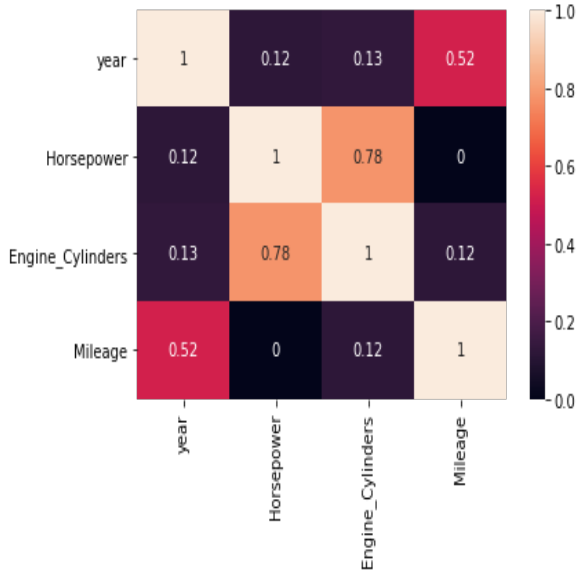
Also we made animated interactive plot showing the connection between price and mileage of each car from 2000 to 2018 where we can differentiate brands with colors of dots.

3.2 Linear Regression

To make our model efficient we wanted to eliminate correlated variables. This shows the correlations between them. From it we can see that the variables horsepower, engine volume and number of cylinders are mutually correlated, moreover, cylinders is present in 2 highest correlations so we removed it. But as we still have a correlation which is higher than 0.8.



Than we tried removing engine volume, which improve the situation



3.3 Fitting the linear model

After removing the engine volume variable, we try fitting the data using statsmodels package's ordinary least squares. As we had a lot of categorical data, we used pandas to create dummy variables. The summary of the fit shows that we have a good fit, as R squared is high and is close to adjusted R square. We use sklearn package to divide our data into train and test, in order to assess the goodness of our fit. Here is the summary.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.900
Model:	OLS	Adj. R-squared:	0.896
Method:	Least Squares	F-statistic:	240.3
Date:	Fri, 16 Aug 2019	Prob (F-statistic):	0.00
Time:	13:33:23	Log-Likelihood:	-67115.
No. Observations:	7462	AIC:	1.348e+05
Df Residuals:	7193	BIC:	1.366e+05
Df Model:	268		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.171e+06	2.19e+04	-53.546	0.000	-1.21e+06	-1.13e+06
year	610.2719	11.332	53.852	0.000	588.057	632.487
Horsepower	19.8997	1.095	18.177	0.000	17.754	22.046
Engine_Cylinders	-105.4643	43.033	-2.451	0.014	-189.821	-21.107
Mileage	-0.0036	0.001	-7.075	0.000	-0.005	-0.003

The first of these plots shows that true values of the price versus the predicted ones are around the line y=x. And the last picture is plot of the errors, and we can see that the distribution is similar to normal, with the mean 0.