Web Scrapping and Linear Regression

Abstract:

The goal of this project is to find a website, get its data, then do EDA after that, finally to do Linear Regression on the data. The website in this project presents a good selection of cars. The site contained data about various types of car, features and cars from different countries: Gulf countries(eg:KSA,...) + Egypt. The site's name is YallaMotor: https://ksa.yallamotor.com/

Design:

The features in a car are many whether the car is new or used. The features can be mentioned in the car description, inferred, filled(based on similar) or simply not there. After studying the site it showed that the used cars had different features, as expected, but also different sellers mention/attach different features, of which some are missing...etc. As for the new cars it had a similar issue but less variant because the sellers are more professional compared to individuals. For the purpose/scope of this project the new car data was scraped.

Data:

The data from the YallaMotor was scrapped using BeautifulSoup and Selenium because the site was using JavaScript. The site had a car list on each page. Each page contained x amount of cars in that last. Usually 15 cars per page. Each country had a different number of pages. The process of scrapping was as follows. First we select a country and get the number of pages it has. Then, starting with the first page, we get the list of cars. By getting the list of cars here it means getting the links related to each of the cars in that list. Once the page is done, it moves to the next page, gets the data, then moves… and so on. Once we have collected all the pages of country C. We change the country and repeat the steps above. Finally once we have collected all the data from all the site. We run the links through the scrape function which extracts the data we want from each page. If the pre-defined feature exists in the page it sets as one if not then zero. The way it was done here is by extracting the data that we want and find. Whatever is missing is then filled with 0 to make all list of each feature the same. All of the different lists of features at the end are combined and turned to a dataframe. The total data was 7700 rows with 81 features. Which were divided into three section: Main, Safety and Fancy. Main included:Engine, Fuel type, Cylinders...etc. Safety included: sensors, headlight type...etc and finally Fancy included, USB, Monitors...etc.

Data cleaning - Major:

The data scraped had some major issues from padding, seeding, dtype...etc. The padding and seeding are caused by how the function was built which assumes -1 at each location to start the process. This seeding process also caused issues with the list length of each feature. Each feature has a list. After adding a feature we check the length of each feature. If its not equal. Then its missing that feature as it was not detected. So a zero is placed in that location to fill the missing value making the length of the list equal. This was done during the scrapping. The data also contained string values: dtype issued str to int/float, white-spaces, squished data(7 seats), nominal data...etc. EDA solved most of those issues.

Algorithms:

The required algorithm is Linear Regression. Which was done in two ways. The first on each collection of features vs price. Then based on correlation.

Tools:

Numpy, Pandas for data manipulation | BeautifulSoup | Selenium | Firefox

Communication:

In addition to the slides and visuals presented, the code is on github bushnag-ai and data is from these sites:

https://ksa.yallamotor.com
https://uae.yallamotor.com
https://kuwait.yallamotor.com
https://bahrain.yallamotor.com
https://qatar.yallamotor.com
https://egypt.yallamotor.com