Supraja Rapuru

Bellevue University

**DSC 540: Data Preparation**

November 17, 2021

As a part of my project deliverables, I have considered three datasets regarding vehicle sales and mileage based on the fuel type of a vehicle. The three datasets are linked based on the Make and Model of a given vehicle.

**Data source as a flat file**, I have chosen a CSV file containing the vehicle demographics, which is specific to the wheelbase, fuel type, and respective price of several motor vehicle models. Each row is one vehicle model, and each column is one specification such as engine size, fuel system, stroke, horsepower, highway mpg, city mpg, etc.

To complete Milestone 2,

1. The exercises for weeks 3 and 4 helped me a lot on how to extract data from CSV files and perform data cleansing techniques. I have completed the following steps to finish up the task.
2. I have renamed few column names to make it more meaningful to understand and deduped the complete dataset.
3. I have subjected the data set to pandas\_profiling to identify the correlation between variables missing values and outliers in the dataset.
4. I have created two new columns, Make and Model, to make data more granular and easy to join in coming milestones.

**Data source as website**, I have chosen one that contains a table of the number of each vehicle model sold during 2020. Each row is one vehicle model. To complete Milestone 3, I have learned and performed the following:

1. The exercises for weeks 4 and 5 helped me a lot on how to extract data from HTML sources and perform data parsing techniques. I have completed the following steps to finish up the task.
2. I will import the HTML data and get it into a more readable format, such as a data frame using BeautifulSoup. I have learned how to use user agents and apply Class to find the keyword-defined values in an HTML tag. It was tricky for me initially, but finally, I have made it.
3. I have removed other redundant data which is not required for my use case. I will separate the make and model names into two columns, just like the CSV.
4. To match the file more manageable, I will only keep the columns for model, sales, and change.
5. The website represents the make and model as car models. I split based on the pattern and assigned the values to respective columns to make the data more granular.

**Data source as API**, I have chosen 2 API datasets to support my use case that contains the vehicle catalog and Vehicle mechanical demographics. To complete Milestone 4, I will conduct the following:

1. I have converted the returned data into JSON format for ease of access.
2. I have renamed and dropped a few of the columns which are not required for my use case.
3. I have organized the retrieved data into a data frame.
4. I have merged two API source data and created a cleansed dataframe with make and model as a grain of the data.

**Milestone 5**, I have loaded individual data from Flatfile, website, API dataframes to separate tables using sqllite. I have encountered a performance issue due to the Cartesian join on API datasoure merge process. However, I have figured the problem and loaded the Final set of data to the API table.

I have created the below visualizations:

* Scatter plot of horsepower (flat file) vs. Engine\_Capacity (website)
* Bar chart of average car Price (API) for each car make (flat file)
* Viewing the sales (website) for each make (flat file) as a pie chart
* viewing Histogram of 'engine\_fuel (API) vs. city mpg (flat-file)
* Density plot of duration\_listed (flat file)
* Boxplots of the Highway mpg distribution (flat file) for each car make (flat file)

**Data Source References:**

CSV: [https://www.kaggle.com/ashydv/car-price-prediction?select=CarPrice\_Assignment.csv](https://www.kaggle.com/ashydv/car-price-prediction?select=CarPrice_Assignment.csv" \t "_blank)

Website: [https://www.carroya.com/buscar/vehiculos/t4e0.do](https://www.carroya.com/buscar/vehiculos/t4e0.do" \t "_blank)

API: [https://parseapi.back4app.com/classes/Carmodels\_Car\_Model\_List?count=1&limit=100](https://parseapi.back4app.com/classes/Carmodels_Car_Model_List?count=1&limit=100" \t "_blank)

Kaggle API: [https://www.kaggle.com/lepchenkov/usedcarscatalog?select=cars.csv](https://www.kaggle.com/lepchenkov/usedcarscatalog?select=cars.csv" \t "_blank)